Version 0.97.2

***A Practical Guide for Developers, Managers, OS Experts, and Companies***

# Open Source License Compendium

## How to Achieve Open Source License Compliance*

Karsten Reincke† Greg Sharpe‡

19th August 2013



†) Deutsche Telekom AG, Products & Innovation, T-Online-Allee 1, 64295 Darmstadt

‡) Deutsche Telekom AG, Telekom Deutschland GmbH, Landgrabenweg, Bonn

*The Open Source Community is a swarm: it is more powerful than a set of arbritarily selected experts. We are proud to have its support. Gladly we thank (in alphabetical order):*

Eitan Adler,
Stefan Altmeyer,
John Dobson,
Steffen Härtlein,
Ta'Id Holmes,
Michael Kern,
Michael Machado,
Thorsten Müller,
Oliver Podebradt,
Thomas Quiehl,
Peter Schichl,
Michael Schierl,
Helene Tamer,
Bernhard Tsai,
. . . additionally all the feedback giving participants of the European Legal & Licensing Workshop 2013 in Amstardam and all the others. . .

# Contents

# Outstanding Parts

- Insert task lists for AGPL, CDDL, MS-RL ...
- Improve / complete the expositions of the concept of a derivative work in the context of software development
- Discuss the license compatibility
- Explain the relationship between open source and earning money
- Complete the discussion of the dynamically and statically linked open source software by a summary of the respective secondary literature
- Improve / expand the integration of the used secondary literature
- Improve the chapter concerning the myths of open source
- Replace / erase the prolegomena appropriately
- Rewrite the BSD discussions at the end of the task list
- Rewirte the MIT discussions at the end of the task list
- Rewrite the PHP discussions at the end of the task list
- Rewrite the PgL discussions at the end of the task list
- Rewrite the Apache discussions at the end of the task list
- Incoporate the secondary literature which has indirectly been used
- Update the literature list

Table 0.1: History of the Open Source License Compendium

| | |
|---|---|
| 2013-08-19 | 0.97.2 ▷ incorpation of the typo fixes offered by M. Schierl▷ enhancing that the OSLiC deals with |
| 2013-07-28 | 0.97.1 |
| 2013-05-20 | 0.96.1 |
| 2013-04-15 | 0.95.2 |
| 2013-04-05 | 0.95.1 |
| 2013-03-15 | 0.94.1 |
| 2013-03-08 | 0.90.1 |
| 2013-02-16 | 0.8.90 |
| 2012-12-28 | 0.8.0 |
| 2012-08-25 | 0.5.2 |
| 2012-07-06 | 0.4.0 |
| 2012-03-22 | 0.2.1 |
| 2012-01-31 | 0.1.8 |
| 2011-09-29 | 0.1.4 |
| 2011-09-12 | 0.1.0 |

# Disclaimer

This book shall be thoroughly developed – together with the open source community. Finally, it shall deliver reliable information based on the insight that the swarm knows more than the single fish.

But nevertheless, the OSLiC can not offer more than the opinion(s) of its authors and contributors. It is only one voice of the chorus discussing the open source licenses. For protecting the authors and contributors from charges and claims of indemnification we adopt the lightly modified GPL3 disclaimer:

THERE IS NO WARRANTY FOR THE OSLiC, TO THE EXTENT PERMITTED BY APPLICABLE LAW. THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE TEXT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE OSLiC IS WITH YOU. SHOULD THE OSLiC PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE OSLiC AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE OSLiC (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE OSLiC TO COOPERATE WITH ANY OTHER TOOLS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

*Particularily, it must be highlighted that - referred to your solitary case - the OSLiC can not and shall not replace a legal review or a legal advice by lawyers. The OSLiC is dealing with with prototypic use cases. It may inspire developers, managers, open source experts, and companies to find good solutions which they finally should let be reviewed by legal counselors*[1].

[1)] For German readers: The OSLiC naturally respects the German 'Rechtsdienstleistungsgesetz'. It only contains legal reflections addressed to a general public. The OSLiC may

only be read as an "nur an die Allgemeinheit gerichtete Darstellung und Erörterung von Rechtsfragen".

# 1 Introduction

*This chapter briefly describes the idea behind the OSLiC, the way it should be used and the way it can be read – which is indeed not quite the same.*

This book focuses on just one issue: *What needs to be done in order to act in accordance with the licenses of those* open source software *we use?* The *Open Source License Compendium* aims at reliably answering this question – in a simple and easy to understand manner. However, it is not just another book on *open source* in general[2]. The intention is, rather, for it to be a tool for simplifying the activities for achieving license conformity.

This compendium was created out of a necessity at *Deutsche Telekom AG* and a challenge for some of its software developers and project managers: Of course, the company itself wants to behave as license compliantly as its employees. Unfortunately, they could not find a reference text which simply lists what precisely must be done in order to comply with the license of that piece of open source being used.

As some of these co-workers in Telekom projects, even we – the initial authors of the OSLiC – did not want to become open source license experts only for being able to use open source software in accordance with the respective licenses. We did not want to become lawyers. We just wanted to do more efficiently, what in those days claimed much time and many resources. We were searching for clear guidance instead of having to determine a correct way through the jungle of open source licenses – over and over again, project for project. We loved using the high-quality open source software to improve our performance. We liked using

[2] Meanwhile, there are tons of literature dealing with open source. Trying to expand your knowledge by means of books and articles might let you get lost in literature: our list of secondary literature may adumbrate this 'danger of being overwhelmed'. But nevertheless, our bibliography at the end of the OSLiC is not complete. Moreover, it is not intended to be complete. It is only an extract representing the background information we did not directly cite in the OSLiC. If we were forced to indicate two books for attaining a good overview on the topic of *open source (licenses)* we would name (a) the 'Rebel Code' (for a German version cf. *Moody, Glyn*: Die Software-Rebellen. Die Erfolgsstory von Linus Torvalds und Linux; transl. from the American [edition, 2000] by Annemarie Pumpering; Landsberg am Lech: verlag moderne industrie, 2001, ISBN 3–478–38730–2, passim – for an English version cf. *Moody, Glyn*: Rebel Code: Linux And The Open Source Revolution; [New York]: Basic Books, 2002, ISBN ISBN 978–0738206707, p. passim) and (b) the 'legal basic conditions' (cf. *Jaeger, Till* a. *Axel Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 3rd edition. München: Verlag C.H. Beck, 2011, passim). But fortunately, we are not forced to do so.

it legally. But we did not like to laboriously discuss the legal constraints of the many and different open source licenses.

What we needed, was an easy-to-use handout which would lead us without any detours to executable lists of work items. We wished to obtain to-do lists, tailored to our usecases and our licenses. We needed reliable lists of tasks we only had to execute for being sure that we were acting in accordance with the open source license. When we started out, such a compendium did not exist.

For solving this problem our company took three decisions:

The first decision our company arrived at, was to support a small group of employees to act as *a board of open source license experts*: They should offer a service for the whole company. Projects, managers, and developers should be able to ask this board what they have to do for complying with a specific open Source License under specific circumstances. And this board should answer with authoritative to-do lists whose executions would assure that the requestors are acting according to the corresponding open source licenses. The idea behind this decision was simple. It would save cost and increase quality if one had one central group of experts instead of being obliged to select (and to train) developers – over and over again, for every new project. So, the *OSRB* – the *Telekom Open Source Review Board* – was founded as an internal expert group – as a self-organizing, bottom-up driven community.

The second decision our company took, was to allow this *Telekom OSRB* to collect their results systematically – in the form of a reusable compendium. The idea behind this decision was also simple: The more the internal service became known, the more the workload would increase: the more work, the more resources, the more costs. So, such a compendium should save costs and enable the requestors to find answers by themselves without becoming license experts: For all default cases, they should find an answer in the compendium instead of having to request that their work is analyzed by the OSRB. Thus, the planned *Telekom Open Source License Compendium* prevents Telekom from being forced to increase the number of OSRB members in the future.

The third decision our company reached, was to allow the *Telekom OSRB* to create the compendium in the same mode of cooperation that open source projects usually use. Again, a simple reason evoked this ruling: If in the future – as a rule – not a reviewing OSRB, but a simple manual should assure the open source license compliant behavior of projects, programmers and managers, this book had of course to be particularly reliable. There is a known feature of the open source working model: the ongoing review by the cooperating community increases the quality. Therefore, the decision, not only to write an internal 'Telekom handout', but to enable the whole community to use, to modify and to redistribute a broader *Open Source License Compendium*, was a decision for improving quality. Consequently, the *OSRB* decided to publish the *OSLiC* as a set of LaTeX

sources, publicly available via the open repository GitHub[3]. And it licensed the OSLiC under Creative Commons Attribution-ShareAlike 3.0 Germany License[4].

But to publish the *OSLiC* as a free book has another important connotation – at least for the *Telekom OSRB*: It is also intended to be an appreciative *giving back* to the *open source community* which has enriched and simplified the life of so many employees and companies over so many years.

Altogether, the OSLiC follows five principles:

**To-do lists as the core, discussions around them** : Based on a simple form to gather information concerning the concrete use of a piece of open source software and its license, the OSLiC shall offer an easy-to-use finder taking the requestor to the respective to-do list for ensuring license conformity. In addition, all these elements of the OSLiC should comprehensibly be introduced and discussed without disturbing the usage itself.

**Quotations with thoroughly specified sources** : The OSLiC shall be revisable and reliable. It shall comprehensibly argue and explicitly specify why it adopts which information, from whom, in which version, and why[5].

**No clearing the forest, but cutting a swath** : The OSLiC has to deal with li-

[3] Get the code by using the link `https://github.com/dtag-dbu/oslic`; get project information by `http://dtag-dbu.github.com/oslic/` or by `http://www.oslic.org/`.

[4] This text is licensed under the Creative Commons Attribution-ShareAlike 3.0 Germany License (`http://creativecommons.org/licenses/by-sa/3.0/de/`): Feel free "to share (to copy, distribute and transmit)" or "to remix (to adapt)" it, if you "[...] distribute the resulting work under the same or similar license to this one" and if you respect how "you must attribute the work in the manner specified by the author(s) [...]"): In an internet based reuse please mention the initial authors in a suitable manner, name their sponsor *Deutsche Telekom AG* and link it to `http://www.telekom.com`. In a paper-like reuse please insert a short hint to `http://www.telekom.com`, to the initial authors, and to their sponsor *Deutsche Telekom AG* into your preface. For normal citations please use the scientific standard.

[5] For that purpose, we are using an 'old-fashioned' bibliographic style with footnotes, instead of endnotes or inline-hints. We want to enable the users to review or to ignore our comments and hints just as they prefer – but on all accounts without being disturbed by large inline comments or frequent page turnings. We know that modern writer guides prefer less 'noisy' styles (pars pro toto cf. *MLA*: MLA Handbook for Writers of Research Papers; 7th edition. New York: The Modern Language Association of America, 2009, ISBN 978–1–60329–024–1, passim). But for a reliable usage – challenged by the often modified internet sources – these methods are still a little imprecise (for details → OSLiC, pp. 211. For a short motivation of the style used in the OSLiC cf. *Reincke, Karsten*: Classical Scholar Texts With Footnotes based on LaTeX, BibTeX, Koma, jurabib and mykeds-CSR; 2012 ⟨URL: `http://www.fodina.de/en/closedprojects/latex-addons/classical-scholar.html`⟩ – reference download: 2013-02-10, passim. For a more elaborated legitimizing version cf. *Reincke, Karsten*: (Geistes-) Wissenschaftliche Texte mit jurabib. Dienst am Leser, Dienst am Scholaren: Uber Anmerkungsapparate in Fußnoten - aber richtig. [n.l.], 2012 ⟨URL: `http://download.fodina.de/fodinaClassicalScholarFoNoDe.pdf`⟩ – reference download: 2013-02-10, passim).

censes and their legal aspects, no doubt. But it shall not discuss all details of every aspect. It shall focus on one possible way to act according to a license in a specific usecase – even if it is known that there might be alternatives[6].

**Take the license text seriously** : The OSLiC shall not give general lectures on legal discussions, much less shall it participate in them. It shall only find one dependable way for each license and each usecase to comply with the license. The main source for this analysis shall be the exact reading of the open source licenses themselves – based on and supported by the interpretation of benevolent lawyers and rationally arguing software developers. The OSLiC shall respect that open source licenses are written for software developers (and sometimes by developers).

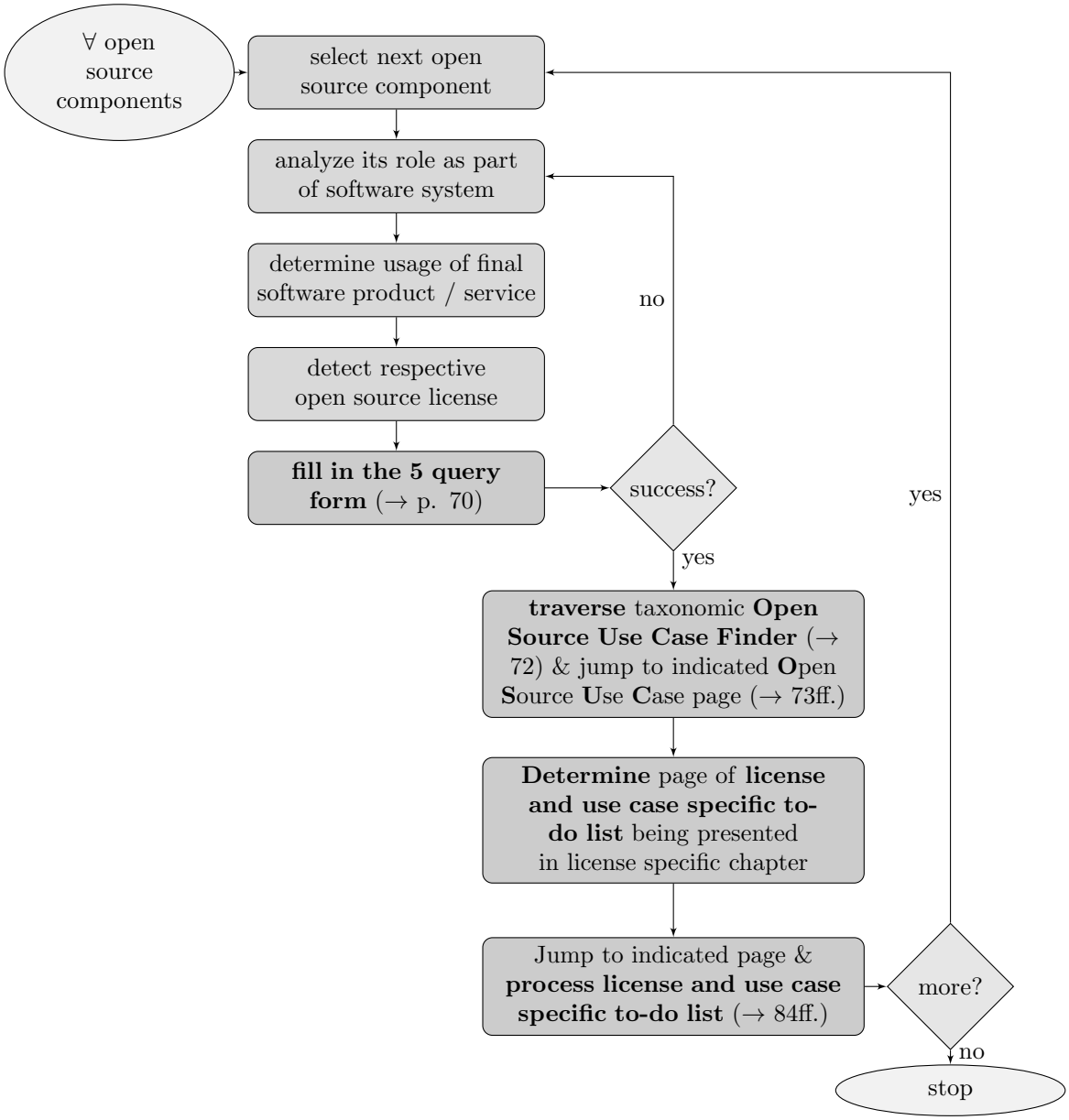
**Trust the swarm** : The OSLiC shall be open for improvements and adjustments encouraged and stimulated also by other people than employees of *Deutsche Telekom AG*.

Based on these principles the OSLiC offers two methods for being used:

First and foremost the readers expect to simply and quickly find those to-do lists fitting their needs. Here is the respective process[7]:

[6] The OSLiC shall not counsel projects with respect to their specific needs. This must remain the task for lawyers and legal experts. The OSLiC cannot and shall not replace a legal review or a legal advice by lawyers. It shall inspire developers, managers, open source experts, and companies to find good solutions which they finally should have reviewed by legal counselors in the end. For the German readers let us repeat again: The OSLiC naturally respects the German 'Rechtsdienstleistungsgesetz'. It only contains legal reflections addressed to a general public. Its content may only be read as a "nur an die Allgemeinheit gerichtete Darstellung und Erörterung von Rechtsfragen".

[7] For the well known 'quick and dirty hackers' – as we tend to be, too – we have integrated a shortcut: If you already know the license of the open source package you want to use and if you are very familiar with the meaning of the open source use cases we defined, then you might directly jump to the corresponding license specific chapter, without 'struggling' with *OSLiC 5 query form* (→ OSLiC p. 70), the taxonomic *Open Source Use Case Finder* (→ 72) or the *O*pen *S*ource *U*se *C*ase page (→ 73ff.): Some of the chapters dedicated to specific open source licenses start with a license specific finder offering a set of license specific use cases – which, according to the complexity of the license, in some cases could be stripped down. But the disadvantage of this method is that you have to apply your knowledge about the use cases and their side effects by yourself without the systematically guiding OSLiC process.




Second, the readers might wish to comprehend the whole analysis. So, we briefly discuss open source licenses taxonomies as the basis for a license compliant behavior[8]. We consider some side effects of acting according to the open source licenses[9]. Finally, we study the structure of open source use cases[10].

So, let us close our introduction by using, modifying, and (re)distributing a well known wish of a well known man: Happy (Legally) Hacking.

[8] → OSLIC "Open Source: The Same Idea, Different Licenses", pp. 16
[9] → OSLiC "Open Source: About Some Side Effects", pp. 47
[10] → OSLiC "Open Source Use Cases: Concept and Taxonomy", pp. 65

# 2 Open Source: The Same Idea, Different Licenses

*This chapter describes different license models which follow the common idea of free open source software. We want to discuss existing ways of grouping licenses to underline the limits of building such clusters: These groups are often used as 'virtual prototypic licenses' which shall deliver a simplified view at the conditions how to act according to the respective real license instances. But one has to meet the requirements of a specific license, not one's own generalized idea of a set of licenses. Nonetheless, also we offer a new structuring view into the world of the open source licenses. We will use a novel set of grouping criteria by referring to the common intended purpose of licenses: each license is designed to protect something or someone against something or someone. Following this pattern, we can indeed summarize all Open Source Licenses in a comparable way.*

Grouping open source licenses is commonly done. Even the set of the *open source licenses*[11] itself is already a cluster being established by a set of grouping criteria: The "distribution terms" of each software license that intends to become an open source license, "[...] must comply with the [...] criteria" of the *Open Source Definition*[12], maintained by the *Open Source Initiative*[13] and often abbreviated as *OSD*. So, this *OSD* demarcates 'the group of [potential] open source licenses' against 'the group of not open sources licenses'[14].

Another way to cluster the *Free Software Licenses* is specified by the "Free Software Definition". This *FSD* contains four conditions which must be met by any free software license: any FSD compliant license must grant "the freedom to run a program, for any purpose [...]", "the freedom to study how it works, and adapt it to (one's) needs [...]", "the freedom to redistribute copies [...]", and finally "the freedom to improve the program, and release your improvements [...]"[15]. Surprisingly this definition implies that the requirement *the sourcecode*

[11] cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted; 2012 [n.y.] ⟨URL: `http://opensource.org/licenses/alphabetical`⟩ – reference download: 2013-01-22, wp.

[12] cf. *Open Source Initiative*: The Open Source Definition; 2012 [n.y.] ⟨URL: `http://www.opensource.org/docs/osd`⟩ – reference download: 2012-06-21, wp.

[13] cf. *Open Source Initiative*: The Open Source Initiative; 2012 [n.y.] ⟨URL: `http://www.opensource.org/about/`⟩ – reference download: 2013-01-22, wp.

[14] More precisely: meeting the OSD is only a necessary condition for becoming an *open source license*. The sufficient condition for becoming an *open source license*, is the approval by the OSI which offers a process for the officially approval of *open source license* (cf. *Open Source Initiative*: The [OSI] Licence Review Process; 2012 [n.y.] ⟨URL: `http://www.opensource.org/approval`⟩ – reference download: 2013-01-22, wp).

[15] cf. *Stallman, Richard M.*: Free Software Definition; originally written in 1996; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, p. 41.

*must be openly accessible*, is 'only' a derived condition. If the "freedom to make changes and the freedom to publish improved versions" shall be "meaningful", then the "access to the source code of the program" is a prerequisite. "Therefore, accessibility of source code is a necessary condition for free software."[16]

The difference between the OSD and the FSD has often been described as a difference of emphasis[17]: Although both definitions "[...] (cover) almost exactly the same range of software", the *Free Software Foundation* – as it is said – "prefers [...] (to emphazise) the idea of freedom [...]" while the *OSI* wants to underline the philosophically indifferent "development methodolgy"[18].

A third method to group of free software and free software licenses is specified by the "Debian Free Software Guideline" which is embedded into the "Debian Social Contract". This "DFSG" contains nine defining criteria which – as Debian itself says – have been "[...] adopted by the free[sic!] software community as the basis of the Open Source Definition"[19].

A rough understanding of these methods might result in the conclusion that these three definitions are extensionally equal and only differ intensionally. But that is not true. To unveil the differences, let us compare the clusters *OSI approved licenses*, *OSD compliant licenses*, *DFSG compliant licenses*, and *FSD compliant licenses* extensionally, by asking whether they *could* establish different sets of licenses[20].

---

16) cf. *Stallman*: Free Software Definition, 1996, p. 41.

17) This is also the viewpoint of Richard M. Stallman: On the one hand, he clearly states that the "Free Software movement" and the "open source movement" generally "[...] disagree on the basic principles, but agree more or less on the practical recommendations" and that he "[...] (does) not think of the open source movement as an enemy". On the other hand, he delineates the two movements by stating that "for the open source movement, the issue of whether software should be open source is a practical question, not an ethical one", while "for the Free Software movement, non-free software is a social problem and free software is the solution" (cf. *Stallman, Richard M.*: Why 'Free Software' is Better than 'Open Software'; originally written in 1998; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, p. 55). Consequently, Richard M. Stallman summarizes the positions in a simple way: "[...] 'open source' was designed not to raise [...] the point that users deserve freedom". But he and his friends want "to spread the idea of freedom" and therefore "[...] stick to the term 'free software'" (id., l.c., p. 59). For a brush-up of this position, expressing again that "(o)pen source is a development methodolgy [and that] free software is a social movement" with an "ethical imparative" cf. *Stallman, Richard*: Viewpoint: Why "Open Source" Misses the Point of Free Software; in: Commununications of the ACM, 52 June (2009), No. 6 ⟨URL: `http://doi.acm.org/10.1145/1516046.1516058`⟩ – reference download: 2011-12-29, p. 31

18) pars pro toto: cf. *Fogel, Karl*: Producing Open Source Software; How to Run a Successful Free Software Project; Beijing, Cambridge, Köln [...]: O'Reilly, 2006, ISBN 978–0–596–00759–1, p. 232.

19) cf. *Debian*: The Debian Free Software Guidelines (DFSG); 2013 [n.y.] ⟨URL: `http://www.debian.org/social_contract#guidelines`⟩ – reference download: 2013-01-22, p. wp.

20) Indeed, for analyzing the extensional power of the definition we have to regard all potentially

First, the difference most easy to determine is that of an unidirectional inclusion: By definition, the *OSI approved licenses* and the *OSD compliant licenses* meet the requirements of the OSD[21]. But only the *OSI approved licenses* have successfully passed the OSI process[22] and therefore are officially listed as *open source licenses*[23]. Hence, on the one hand, *OSI approved licenses* are *open source licenses* and vice versa. On the other hand, both – the *OSI approved licenses* and the *open source licenses* – are *OSD compliant licenses*, but not vice versa.

Second, a similar argumentation allows to distinguish the *DFSG compliant licenses* from the *OSI approved licenses*. As it is stated, the OSD "[...] is based on the Debian Free Software Guideline and any license that meets one definition almost meets the other"[24]. But then again, meeting the definition is not enough for being an official open source license: the license has to be approved by the OSI[25]. Thus, it follows that all *OSI approved licenses* are also *DFSG compliant licenses*, but not vice versa.

Third, – by ignoring the "few exceptions" which have appeared "over the years"[26] – it can be said that, because of their 'kinsmanlike' relation, at least the *OSD compliant licenses* are also *DFSG compliant licenses* and vice versa.

Last but not least, it must be stated that the (potential) set of free software licenses must be greater than all the other three sets: On the one side, the FSD requires that a license of free software must not only allow to read the software, but must also permit to use, to modify, and to distribute it[27]. These conditions are covered by at least the first three paragraphs of the OSD concerning the topics "Free Redistribution", "Source Code", and "Derived Works"[28]. On the other side, the OSD contains at least some requirements which are not mentioned by the FSD and which nevertheless must be met by a license in order to be qualified as an OSD compliant license[29]. It follows then that there may exist licenses which fulfill all conditions of the FSD and nevertheless do not fulfill at least some conditions of the OSD[30]. So, the set of all (potential) *Free Software Licenses* must

covered licenses, not only the already existing licenses, because the subset of really existing licenses still could be expanded be developing new licenses which fit the definition.

21) cf. *Open Source Initiative*: The Open Source Definition, 2012, wp.

22) cf. id., ibid.

23) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

24) cf. *Fogel*: Producing Open Source Software, 2006, p. 233.

25) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

26) cf. *Fogel*: Producing Open Source Software, 2006, p. 233.

27) cf. *Stallman*: Free Software Definition, 1996, p. 41.

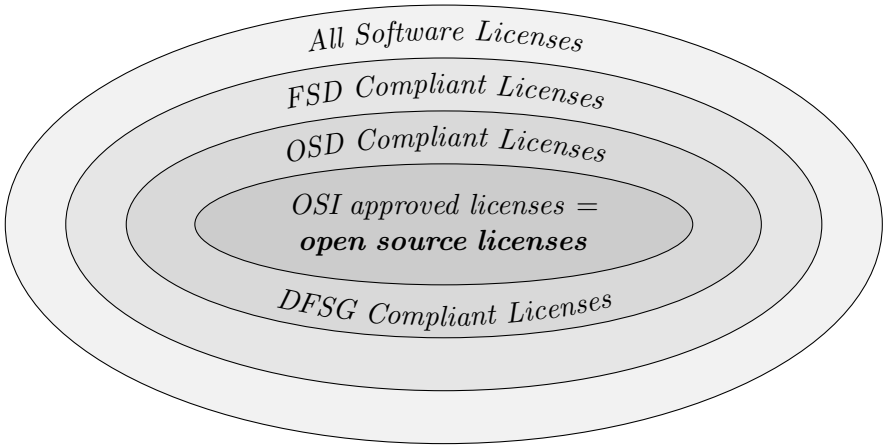
28) cf. *Open Source Initiative*: The Open Source Definition, 2012, wp.

29) For example, see the condition that "the license must be technology-neutral" (cf. id., ibid.).

30) Again: we must consider the extensional potential of the definitions, not the set of really existing licenses. In this context, it is irrelevant that actually all existing Free Software Licenses like GPL, LGPL or AGPL indeed are also classfied as open source licenses. We are referring to the fact that there might be generated licenses which fulfill the FSD, but not

be greater than the set of all (potential) *open source licenses* and greater than the set of *OSD compliant licenses*.

All in all, we can visualize the situation as follows:




It should be clear without longer explanations that these clusters don't allow to extrapolare to the correct compliant behaviour according to the *open source licenses*: On the one hand, all larger clusters do not talk about the *open source licenses*. On the other hand, the *open source license cluster* itself only collects its elements on the base of the OSD which does not stipulates concrete license fulfilling actions for the licensee.

The next level of clustering *open source licenses* concerns the inner structure of these *OSI approved licenses*. Even the OSI itself has recently discussed whether a better kind of grouping the listed licenses would better fit the needs of the visitors of the OSI site[31]. And finally the OSI ends up in the categories "popular and widely used (licenses) or with strong communities", "special purpose licenses", "other/miscellaneous licenses", "licenses that are redundant with more popular licenses", "non-reusable licenses","superseded licenses", "licenses that have been voluntarily retired", and " uncategorized licenses"[32].

Another way to structure the field of open source licenses is to think in "types of open source licenses" by grouping the "*academic licenses*, named as such because they were originally created by academic institutions"[33], the "*reciprocal licenses*",

the OSD.

[31] cf. *Open Source Initiative*: OSI Mailing List. License-discuss. Draft of new OSI licenses landing page; 2012 [n.y.] ⟨URL: `http://projects.opensource.org/pipermail/license-discuss/2012-April/000332.html`⟩ – reference download: 2013-01-29, wp.

[32] cf. *Open Source Initiative*: Open Source Licenses by Category; 2013 [n.y.] ⟨URL: `http://opensource.org/licenses/category`⟩ – reference download: 2013-01-29, wp.

[33] cf. *Rosen, Lawrence*: Open Source Licensing. Software Freedom and Intellectual Property Law; Upper Saddle River, New Jersey: Prentice Hall PTr, 2005, ISBN 0–13–148787–6, p. 69.

named as such because they "[...] require the distributors of derivative works to distribute those works under same license including the requirement that the source code of those derivative works be published"[34], the "*standard licenses*", named as such because they refer to the reusability of "industry standards"[35], and the "*content licenses*", named as such because they refer to "[...] other than software, such as music art, film, literary works" and so on[36].

Both kinds of taxonomy directly help to find the relevant licenses which should be used for new (software) projects. But again: none of these categories allow to infer the license compliant behaviour, because the categories are mostly defined based on license external criteria: whether a license is published by a specific kind of organization or whether a license deals with industry standards or other kind of works than software, inherently do not evoke a license fulfilling behaviour.

Only the act of grouping into the "*academic licenses*" and the "*reciprocal licenses*" touches the idea of license fulfilling doings, if one – as it has been done – expands the definition of the "*academic licenses*" by the specification that these licenses "[...] allow the software to be used for any purpose whatsoever with no obligation on the part of the licensee to distribute the source code of derivative works"[37]. With respect to this additional specification, the clusters "*academic licenses*" and the "*reciprocal licenses*" indeed might be referred as the "main categories" of (open source) licenses[38]: By definition, they are constituting not only a contrary, but contradictory opposite. However, it must be kept in mind that they constitute an inherent antagonism, an antinomy inside of the set of open source licenses[39].

Connatural to the clustering into *academic licenses* and *reciprocal licenses* is the grouping into *permissive licenses*, *weak copyleft licenses*, and *strong copyleft licenses*: Even Wikipedia already uses the term "permissive free software licence" in the meaning of "a class of free software licence[s] with minimal requirements about how the software can be redistributed" and "contrasts" them with the "copyleft licences" as those "with reciprocity / share-alike requirements"[40].

---

[34] cf. *Rosen*: Open Source Licensing, 2005, p. 70.

[35] cf. id., ibid.

[36] cf. id., l.c., p. 71.

[37] cf. id., ibid.

[38] cf. id., l.c., p. 179.

[39] Hence, it is at least a little confusing to say that "the open source license (OSL) is a reciprocal license" and "the Academic Free License (AFL) is the exact same license without the reciprocity provisions" (cf. id., l.c., p. 180): If the BSD license is an AFL and if an AFL is not an OSL and if the OSI approves only OSLs, then the BSD license can not be an approved open source license. But in fact, it still is (cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp).

[40] cf. *Wikipedia (en)*: Permissive free software licence; n.l., 2013 [n.y.] ⟨URL: `http://en.wikipedia.org/wiki/Permissive_free_software_licence`⟩ – reference download: 2013-02-02, wp.

Some other authors name the set of *academic licenses* the "permissive licenses" and specify the *reciprocal licenses* as "restrictive licenses", because in this case – as a consequence of the embedded "copyleft" effect – the source code must be published in case of modifications. They additionally introduce the subset of "strong restrictive licenses" which additionally require that an (overarching) derivative work must be published under the same license[41]. The next refinement of such clustering concepts directly uses the categories "[open source] licenses with a strict copyleft clause"[42], "[open source] licenses with a restricted copyleft clause"[43], and "[open source] licenses without any copyleft clause"[44]. Finally, this viewpoint can directly be mapped to the categories *strong copyleft* and *weak copyleft*: While on the one hand, "only changes to the weak-copylefted software itself become subject to the copyleft provisions of such a license, [and] not changes to the software that links to it", on the other hand, the "strong copyleft" states "[...] that the copyleft provisions can be efficiently imposed on all kinds of derived works"[45].

Based on this approach to an adequate clustering and labeling[46], we can develop the following picture:

[41] pars pro toto cf. *Buchtala, Rouven*: Determinanten der Open Source Software-Lizenzwahl. Eine spieltheoretische Analyse; Frankfurt am Main, Berlin, Bern [... etc.]: Peter Lang, 2007 (= Informationsmanagement und strategische Unternehmensführung), [Vol./No.] 12), ISBN 978–3–631–57114–9, p. 57.

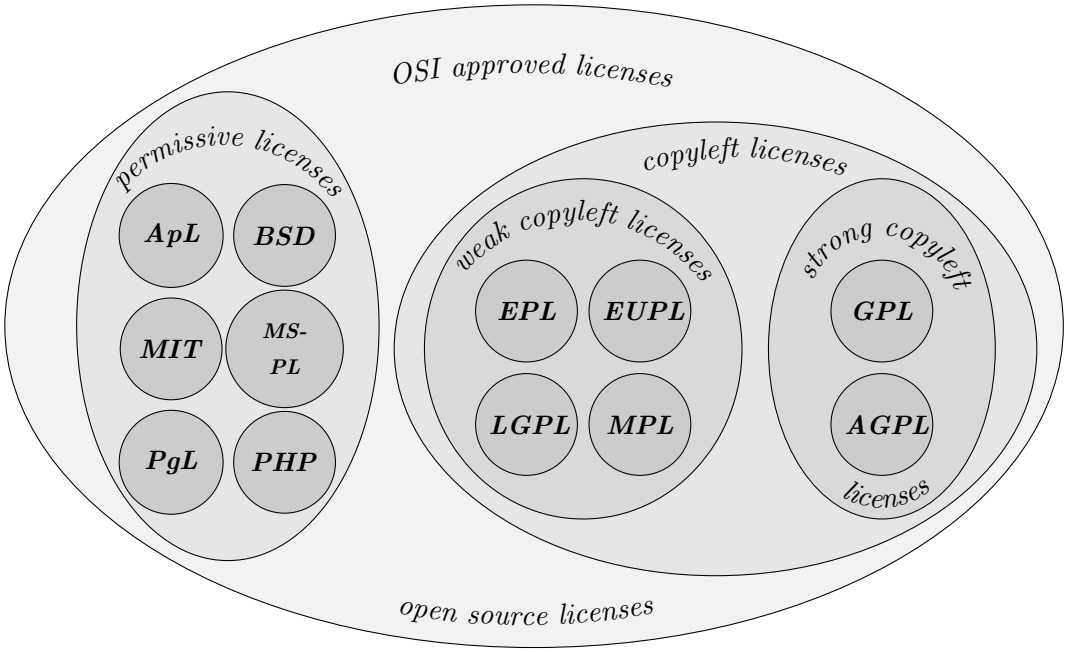
[42] Originally stated as "Lizenzen mit einer strengen Copyleft-Klausel". Cf. *Jaeger a. Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software, 2011, p. 24.

[43] Originally stated as "Lizenzen mit einer beschränkten Copyleft-Klausel". Cf. id., l.c., p. 71.

[44] Originally stated as "Lizenzen ohne Copyleft-Klausel". Cf. id., l.c., p. 83.

[45] cf. *Wikipedia (en)*: Copyleft; n.l., 2013 [n.y.] ⟨URL: `http://en.wikipedia.org/wiki/Copyleft`⟩ – reference download: 2013-02-02, wp.

[46] Finally, we should also mention that there still exists other classifications which might become important in other contexts. For example, the ifross license subsumes under the main category "Open Source Licenses" the subcategories "Licenses without Copyleft Effect", "Licenses with Strong Copyleft", "Licenses with Restricted Copyleft", "Licenses with Restricted Choice", or "Licenses with Privileges" – and let finally denote these categories also licenses which are not listed by the OSI (cf. *ifross*: License Center; 2011 [n.y.] ⟨URL: `http://www.ifross.org/ifross_html/lizenzcenter-en.html`⟩ – reference download: 2013-02-26, wp). This is well reasonable if one refers to the meaning of the OSD (cf. *Open Source Initiative*: The Open Source Definition, 2012, wp). The OSLiC wants to simplify its object of study by referring to the approved open source licenses (cf. *Open Source Initiative*: The [OSI] Licence Review Process, 2012, wp) listed by the OSI (cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp).




This extensionally based clarification of a possible open source license taxonomy is probably well-known and often – more or less explicitly – referred to[47]. Unfortunately, this taxonomy still contains some misleading underlying messages:

*Permissive* has a very positive connotation. So, the antinomy of *permissive licenses* versus *copyleft licenses* implicitly signals, that the *permissive licenses* are in any meaning better, than the *copyleft licenses*. Naturally, this 'conclusion' is evoked by confusing the extensional definition and the intensional power of the labels. But that is the way we – the human beings – like to think.

Anyway, this underlying message is not necessarily 'wrong'. It might be convenient for those people or companies who only want to use open source software without being restricted by the *obligation to give something back* as it has been introduced by the 'copyleft'[48]. But there might be other people and companies who emphasize the protecting effect of the copyleft licenses. And indeed,

[47] Even the FSF itself uses the term 'permissive non-copyleft free software license' (pars pro toto: cf. *Free Software Foundation*: Various Licenses and Comments about Them; 2013 [n.y.] ⟨URL: `http://www.gnu.org/licenses/license-list.html`⟩ – reference download: 2013-02-08, wp/section 'Original BSD license') and contrasts it with the terms 'weak copyleft' and 'strong copyleft' (pars pro toto: cf. id., l.c., wp/section 'European Union Public License')

[48] De facto, *copyleft* is not *copyleft*. Apart from the definition, its effect depends on the particular licenses which determine the conditions for applying the copyleft 'method'. For example, in the GPL, the copyleft effect is bound to the criteria 'being distributed'. Later on, we will collect these conditions systematically (see chapter *Open Source Use Cases: Concept and Taxonomy*, pp. 65). Therefore, here we still permit ourselves to use a somewhat 'generalizing' mode of speaking.

at least the open source license[49] *GPL*[50] has initially been developed to protect the freedom, to enable the developers to help their "neighbours" and to get the modifications back[51]: So, "Copyleft" is defined as a "[...] method for making a program free software and requiring all modified and extended versions of the program to be free software as well"[52]. It is a method[53] by which "[...] the code and the freedoms become legally inseparable"[54]. Because of these disparate interests of hoping not to be restricted and hoping to be protected, it could be helpful to find a better label – an impartial name for the cluster of *permissive licenses*. But until that time, we should at least know that this taxonomy still contains an underlying declassing message.

The other misleading interpretation is – counter-intuitively – evoked by using the concept of 'copyleft licenses'. By referring to a cluster of *copyleft licenses* as the opposite of the *permissive licenses*, one implicitly also sends two messages: First, that republishing one's own modifications is sufficient to comply with the *copyleft licenses*. And, secondly, that the *permissive licenses* do not require anything to be done for obtaining the right to use the software. Even if one does not wish to evoke such an interpretation, we – the human beings – tend to take the things as simple as possible[55]. But because of several aspects, this understanding of the

[49] Although RMS naturally prefers to specify it as a *Free Software License* (s. p. 17)

[50] As the original source cf. *Free Software Foundation*: GNU General Public License, version 2; 1991 [n.y. of the html page itself] ⟨URL: `http://www.gnu.org/licenses/gpl-2.0.html`⟩ – reference download: 2013-02-05, wp. Inside of the OSLiC, we constantly refer to the license versions which are published by the OSI, because we are dealing with officially approved open source licenses. For the 'OSI-GPL' cf. *Open Source Initiative*: GNU General Public License, version 2 (GPL-2.0). Version 2, June 1991; 1991 [n.y. of the html page itself] ⟨URL: `http://opensource.org/licenses/GPL-2.0`⟩ – reference download: 2013-02-05, wp

[51] The history of the GNU project is multiply told. For the GNU project and its initiator cf. pars pro toto *Williams, Sam*: Free as in Freedom. Richard Stallman's Crusade for Free Software; Beijing [... etc.]: O'Reilly, 2002, ISBN 0–596–00287–4, passim. For a broader survey cf. pars pro toto *Moody*: Die Software-Rebellen, 2001, passim. A very short version is delivered by Richard M. Stallman himself where he states that – in the years while the early free community were destroyed – he saw the "nondisclosure agreement" which must be signed , "[...] even to get an executable copy" as a clear "[...] promise not to help your neighbour": "A cooperating community was forbidden." (cf. *Stallman, Richard M.*: The GNU Project; originally published in 'Open Sources: Voices from the Open Source Revolution, O'Reilly, 1999'; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, p. 16).

[52] cf. *Stallman, Richard M.*: What is Copyleft? originally written in 1996; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, p. 89.

[53] Based on the American legal copyright system, this method uses two steps: firstly one states, "[...] that it is copyrighted [...]" and secondly one adds those "[...] distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code or any program derived from it but only if the distribution terms are unchanged" (cf. id., ibid.).

[54] cf. id., ibid.

[55] And indeed, in the experience of the authors – sometimes – such simplifications gain their

antinomy of *copyleft licenses* and *permissive licenses* is too misleading for taking it as a serious generalization:

On the one hand, even the 'strongly copylefted' GPL requires also other tasks than just the republishing of derivative works. For example, it also calls for to "[...] give any other recipients of the [GPL licensed] Program a copy of this License along with the Program"[56]. Furthermore, the 'weakly copylefted' licenses require also more and different criteria which have to be fulfilled for acting according to these licenses. For example, the EUPL requires that the licensor who does not directly deliver the binaries together with the sourcecode, must offer a sourcecode version of his work free of charge[57], while the MPL requires that under the same circumstances a recipient "[...] can obtain a copy of such Source Code Form [...] at a charge no more than the cost of distribution to the recipient [...]"[58]. And last but not least, also the *permissive licenses* require tasks which must be fulfilled for a license compliant usage – moreover, they also require different things. For example, the BSD demands that "the (re)distributions [...] must (retain [and/or]) reproduce the above copyright notice [...]". Because of the structure of the "copyright notice", this compulsory notice implies that the authors / copyright holders of the software must be publicly named[59]. As opposed to this, the Apache License requires that "if the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file" which often means that you have to present central parts of such files publicly[60] – parts which can contain much more information than only the

independent existence and determine decisions on the management level. But that is not the fault of the managers. It is their job, to aggregate, generalize and simplify information. It is the job of the experts, to offer better viewpoints without overwhelming the others with details.

56) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §1.

57) The German version of the EUPL uses the phrase "problemlos und unentgeltlich(sic!) auf den Quellcode (zugreifen können)" (cf. *Europäische Gemeinschaft* a. *European commission Joinup*: Open-Source-Lizenz für die Europäische Union; 2007 ⟨URL: `http://joinup.ec.europa.eu/system/files/DE/EUPL%20v.1.1%20-%20Lizenz.pdf`⟩ – reference download: 2013-02-08, pp. 3, section 3) while the English version contains the specification "the Source Code is easily and freely accessible" (cf. *European Community* a. *European commission Joinup*: European Union Public Licence v. 1.1. 2007 ⟨URL: `http://joinup.ec.europa.eu/system/files/EN/EUPL%20v.1.1%20-%20Licence.pdf`⟩ – reference download: 2013-02-08, pp. 2, section 3)

58) cf. *Open Source Initiative*: Mozilla Public License 2.0 (MPL-2.0); 2013 [n.y.] ⟨URL: `http://opensource.org/licenses/MPL-2.0`⟩ – reference download: 2013-02-07, section 3.2.a.

59) cf. *Open Source Initiative*: The BSD 2-Clause License; 2012 [n.y.] ⟨URL: `http://www.opensource.org/licenses/BSD-2-Clause`⟩ – reference download: 2012-07-03, wp.

60) cf. *Open Source Initiative*: Apache License, Version 2.0; 2004 [n.y. of the page itself] ⟨URL: `http://opensource.org/licenses/Apache-2.0`⟩ – reference download: 2013-02-07, wp. section 4.4.

names of the authors / copyright holders.

So, no doubt – and contrary to the intuitive interpretation of this taxonomy – each *open source license* must be fulfilled by some actions, even the most permissive one. And for ascertaining these tasks, one has to look into these licenses themselves, not the generalized concepts of licenses taxonomies. Hence again, we have to state that even this well known type of grouping of *open source licenses* does not allow to derive a specific license compliant behavior: The taxonomy might be appropriate, if one wants to live with the implicit messages and generalizations of some of its concepts. But the taxonomy is not an adequate tool to determine, what one has to do for fulfilling an *open source license*. A license compliant behaviour for obtaining the right to use a specific piece of *open source software* must be based on the concrete *open source license* by which the licensor has licensed the software. There is no shortcut.

Nevertheless, human beings need generalizing and structuring viewpoints for enabling themselves to talk about a domain – even if they finally have to regard the single objects of the domain for specific purposes. We think that there is a subtler method to regard and to structure the domain of *open source licenses*. So, we want to offer this other possibility to cluster the *open source licenses*[61]:

We think that, in general, licenses have a common purpose: they should protect someone or something against something. The structure of this task is based on the nature of the word 'protect' which is a trivalent verb: it links someone or something who protects, to someone or something who is protected and both combined to something against which the protector protects and against the other one is protected. Licenses in general do that. Moreover, to "protect" the "rights" of the licensees is explicitly mentioned in the GPL-2.0[62], in the LGPL-2.1[63], and the GPL-3.0[64] – by which the LGPL-3.0 inherits this purpose[65]. Following this viewpoint, we want to generally assume that open source licenses are designed to protect: They can protect the user (recipient) of the software, its contributor resp. developer and/or distributor, and the software itself. And they can protect them against different threats:

- First, we assume, that – in the context of open source software – the user

[61] even if also we have to concede that, ultimately, one has to always look into the license itself

[62] cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. Preamble.

[63] cf. *Open Source Initiative*: The GNU Lesser General Public License, version 2.1 (LGPL-2.1); 1999 [n.y. of the html page itself] ⟨URL: `http://opensource.org/licenses/LGPL-2.1`⟩ – reference download: 2013-03-06, wp. Preamble.

[64] cf. *Open Source Initiative*: GNU General Public License, version 3 (GPL-3.0); 2007 [n.y. of the html page itself] ⟨URL: `http://opensource.org/licenses/GPL-3.0`⟩ – reference download: 2013-03-05, wp. Preamble.

[65] cf. *Open Source Initiative*: The GNU Lesser General Public License, version 3.0 (LGPL-3.0); 2007 [n.y. of the html page itself] ⟨URL: `http://opensource.org/licenses/LGPL-3.0`⟩ – reference download: 2013-03-06, wp. prefix.

can be protected against the loss of the right to use it, to modify it, and to redistribute it. Additionally, he can be protected against patent disputes.

- Second, we assume, that open source contributors and distributors can be protected against the loss of feedback in the form of code improvements and derivatives, against warranty claims, and against patent disputes.
- Third, we assume, that the open source programs and their specific forms – may they be distributed or not, may they be modified or not, may they be distributed as binaries or as sources – can be protected against the re-closing resp. against the re-privatizing of their further development.
- Fourth, we want to assume that new on-top developments being based on open source components can be protected against the privatizing for enlarging the world of freely usable software[66].

With respect to these viewpoints, one gets a subtler picture of the license specific protecting power. Thus, we are going to describe and deduce the protecting power of each of the open source licenses on the following pages. Table 2.1 summarizes the results as a quick reference[67].

## 2.1 The protecting power of the GNU Affero General Public License (AGPL) [tbd]

- The GNU Affero General Public License protects ...
- But the GNU Affero General Public License does not protect ...

## 2.2 The protecting power of the Apache License (ApL)

As an approved *open source license*[68], the Apache License[69] protects the user against the loss of the right to use, to modify and/or to distribute the received

[66] In a more rigid version, this capability of a license could also be identified as the power to protect the community against a stagnation of the set of open source software – but this description is at least a little to long to be used by the following pages

[67] → table 2.1 on p. 27

[68] cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

[69] The Apache License, version 2.0 is maintained by the Apache Software Foundation (cf. *Apache Software Foundation*: Apache License, Version 2.0; 2004 ⟨URL: `http://www.apache.org/licenses/LICENSE-2.0`⟩ – reference download: 2011-08-31, wp). Of course, also the OSI is hosting a duplicate of the Apache license (cf. *Open Source Initiative*: APL-2.0, 2004, wp) and is listing it as an officially approved open source license (cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp). The Apache license 1.1 is classified by the OSI as "superseded license"(cf. *Open Source Initiative*: The Open Source Licenses by Category, 2013, wp). In the same spirit, also the Apache Software Foun-

Table 2.1: Open Source Licenses as Protectors

| *Open Source Licenses*[a] | | *are protecting* | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Users** | | | | **Contributors** (Distributors) | | | **Open Source Software** | | | | | **On-Top Develop.** |
| | | | | | | | | | not distributed | distributed as | | | | |
| | | | | | | | | | | unmodified | | modified | | |
| | | *who have already got sources or binaries* | | | | *who spread open source software* | | | | sources | binaries | sources | binaries | |
| | | *against* | | | | | | | | | | | | |
| | | the loss of the right to | | | Patent Disputes | Loss of Feedback | Warranty Claims | Patent Disputes | Re-Closings / Re-Privatizings of already opened software | | | | | Privatizings |
| | | use it | modify it | redistribute it | | | | | | | | | | |
| ApL | 2.0 | ✓ | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ¬ | ✓ | ¬ | ✓ | ¬ | ¬ |
| BSD | 3-Cl | ✓ | ✓ | ✓ | ¬ | ¬ | ✓ | ¬ | ¬ | ✓ | ¬ | ✓ | ¬ | ¬ |
| | 2-Cl | ✓ | ✓ | ✓ | ¬ | ¬ | ✓ | ¬ | ¬ | ✓ | ¬ | ✓ | ¬ | ¬ |
| MIT | | ✓ | ✓ | ✓ | ¬ | ¬ | ✓ | ¬ | ¬ | ✓ | ¬ | ✓ | ¬ | ¬ |
| MS-PL | | ✓ | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ¬ | ✓ | ¬ | ✓ | ¬ | ¬ |
| PgL | | ✓ | ✓ | ✓ | ¬ | ¬ | ✓ | ¬ | ¬ | ✓ | ¬ | ✓ | ¬ | ¬ |
| PHP | 3.0 | ✓ | ✓ | ✓ | ¬ | ¬ | ✓ | ¬ | ¬ | ✓ | ¬ | ✓ | ¬ | ¬ |
| CDDL | 1.0 | ✓ | ✓ | ✓ | – | – | – | – | – | – | – | – | – | – |
| EPL | 1.0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ✓ | ✓ | ¬ |
| EUPL | 1.1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ✓ | ✓ | ¬ |
| LGPL | 2.1 | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ¬ | ¬ | ✓ | ✓ | ✓ | ✓ | ¬ |
| | 3.0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ✓ | ✓ | ¬ |
| MPL | 2.0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ✓ | ✓ | ¬ |
| *MS-RL* | | ✓ | ✓ | ✓ | – | – | – | – | – | – | – | – | – | – |
| AGPL | 3.0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GPL | 2.1 | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ¬ | ¬ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | 3.0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ¬ | ✓ | ✓ | ✓ | ✓ | ✓ |

[a] '✓' indicates that the license protects with respect to the meaning of the column, '¬' indicates that the license does not protect with regard to the meaning of the column, and '–' indicates, that the corresponding statement must still be evaluated. *Slanted names of licenses* indicate that these licenses are only listed in this table while the corresponding mindmap (→ p. 46) does not cover them

copy of the source code or the binaries[70]. Furthermore, based on its patent clause[71], the ApL protects the users against patent disputes[72]. Because of this patent clause and of its "disclaimer of warranty" together with its "limitation of liability", the Apache license also protects the contributors / distributors against patents disputes and warranty claims[73]. Finally, the ApL protects the distributed sources themselves *against* a change of the license which would *reset* the work *as closed software*, because first, one "[...] must give any other recipients of the work or derivative works a copy of (the Apache) license", second, "in the source form of any derivative works that (one) distributes", one has "[...] to retain [...] all copyright, patent, trademark, and attribution notices [...]", and third, one must "[...] include a readable copy [...of the] NOTICE file" being supplied by the original package one has received[74].

But the Apache License does not protect the contributors against the loss of feedback because it does not 'copyleft' the software: the Apache license does not contain any sentence requiring that one has also to publish the source code. In the same spirit, the ApL does not protect the undistributed software or the distributed binaries against re-closings – neither in unmodified nor in modified form – because the Apache License allows to (re)distribute the binaries without also supplying the sources – even if the binaries rest upon sources modified by the distributor. Finally, the ApL does not protect the on-top developments against a privatizing.

## 2.3 The protecting power of the BSD licenses

As approved *open source licenses*[75], the BSD Licenses[76] protect the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[77]. Additionally, they protect the contribu-

dation itself classifies the releases 1.0 and 1.1 as "historic" (cf. *Apache Software Foundation*: Licenses; 2013 [n.y ] ⟨URL: `http://www.apache.org/licenses/`⟩ – reference download: 2013-02-25, wp). Thus, the OSLiC only focuses on the most recent APL-2.0 version. For those, who have to fulfill these earlier Apache licenses it could be helpful to read them as siblings of the BSD-2CL and BSD-3CL licenses.

70) cf. *Open Source Initiative*: APL-2.0, 2004, wp. §2.

71) → OSLiC pp. 51

72) cf. id., l.c., wp. §3.

73) cf. id., l.c., wp. §3, §7, §8.

74) cf. id., l.c., wp. §4.

75) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

76) BSD has to be resolved as *Berkely Software Distribution*. For details of the BSD license release and namings cf. *Open Source Initiative*: The BSD 3-Clause License; 2012 [n.y.] ⟨URL: `http://www.opensource.org/licenses/BSD-3-Clause`⟩ – reference download: 2012-07-04, wp. editorial

77) cf. *Open Source Initiative*: The Open Source Definition, 2012, wp. §1ff.

tors and/or distributors against warranty claims of the software users, because these licenses contain a 'No Warranty Clause'[78]. And finally they protect the distributed sources against a change of the license which closes the sources, because each modification and "redistributions of [the] source code must retain the [...] copyright notice, this list of conditions and the [...] disclaimer"[79]: Therefore it is incorrect to distribute a BSD licensed code under another license – regardless, whether it closes the sources or not[80].

But the BSD Licenses protect neither the users nor the contributors and/or distributors against patent disputes (because they do not contain any patent clause). They do not protect the contributors against the loss of feedback (because they do not 'copyleft' the software). Moreover, they do not protect the undistributed software or the distributed binaries against re-closings – neither in unmodified nor in modified form – because they allow to redistribute only the binaries without also supplying the source code[81]. Finally, the BSD licenses do not protect the on-top developments against a privatizing.

## 2.4 The protecting power of the CDDL [tbd]

As an approved *open source license*[82], the Common Develop and Distribution License protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[83]

[...]

78) one for all version cf. *Open Source Initiative*: The BSD 2-Clause License, 2012, wp.

79) cf. id., ibid.

80) In common sense based discussions you may have heard that BSD licenses allow to republish the work under another, an own license. Taking the words of the BSD License seriously that is not valid under all circumstances: Yes, it is true, you are not required to redistribute the sourcecode of a modified (derivative) work. You are allowed to modify a received version and to distribute the results only as binary code and to keep your improvements closed. But if you distribute the source code of your modifications, you have retain the licensing, because "Redistribution [...] in source [...], with or without modification, are permitted provided that [...] (the) redistributions of source code [...] retain the above copyright notice, this list of conditions and the following disclaimer" (cf. id., ibid.)

81) see both, the BSD-2CL License (cf. id., ibid.), and the BSD-3CL License (cf. *Open Source Initiative*: The BSD 3-Clause License, 2012, wp)

82) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

83) cf. *Open Source Initiative*: Common Development and Distribution License (CDDL-1.0); 2004 [n.y. of the html page itself] ⟨URL: `http://opensource.org/licenses/CDDL-1.0`⟩ – reference download: 2013-04-19, wp. §?.

## 2.5 The protecting power of the Eclipse Public License (EPL)

As an approved *open source license*[84], the Eclipse Public License[85] protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[86]. Furthermore, based on its patent clause[87], the EPL protects the users also against patent disputes[88]. Besides this patent clause, the EPL contains the sections "no warranty" and "disclaimer of liability"[89]. These three elements together protect the contributors / distributors against patents disputes and warranty claims. Finally, the EPL protects the distributed sources themselves *against* a change of the license which would *reset* the work *as closed software*: First, the Eclipse Public Licenses requires that if a work – released under the EPL – "[...] is made available in source code form [...] (then) it must be made available under this (EPL) agreement, too" while this act of 'making avalaible' "must" incorporate a "copy" of the EPL into "each copy of the [distributed] program" or program package[90]. But in opposite to the permissive licenses, the EPL does not only protect the distributed source code – regardless whether it is modified or not. The EPL also protects the distributed modified or unmodified binaries: The EPL allows each modifying "contributor" and distributor "[...] to distribute the Program in object code form under (one's) own license agreement [...]" provided this license clearly states that the "source code for the Program is available" and where the "licensees" can "[...] obtain it in a reasonable manner on or through a medium customarily used for software exchange"[91]. Thus, one has to conclude that the EPL is a copyleft license.

But the Eclipse Public License is not a license with strong copyleft; the EPL uses 'only' a weak copyleft effect[92]: Indeed, the EPL says that for each EPL licensed "program" – distributed in object form – a place must be made known where one can get the corresponding source code[93]. The term 'Program' is defined as any

[84] cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

[85] The Eclipse Public License, version 1.0 is maintained by the Eclipse Software Foundation (cf. *Eclipse Foundation*: Eclipse Public License, Version 1.0; 2005 [n.y. of the page itself] ⟨URL: `http://www.eclipse.org/org/documents/epl-v10.php`⟩ – reference download: 2013-02-20, wp). Of course, also the OSI is hosting a duplicate (cf. *Open Source Initiative*: Eclipse Public License, Version 1.0; 2005 [n.y. of the page itself] ⟨URL: `http://opensource.org/licenses/EPL-1.0`⟩ – reference download: 2013-02-20, wp).

[86] cf. id., l.c., wp. §2a.

[87] → OSLiC pp. 52

[88] cf. id., l.c., wp. §2b & §2c.

[89] cf. id., l.c., wp. §5 & §6.

[90] cf. id., l.c., wp. §3.

[91] cf. id., l.c., wp. §3, esp. §3.b.iv.

[92] Even if one can find contrary specifications in the internet. Pars pro toto cf. *ifross*: ifross Lizenz-Center, 2011, wp: This page is listing the EPL in the section "Other Licenses with strong Copyleft Effect"

[93] cf. *Open Source Initiative*: EPL-1.0, 2005, wp. §3, esp. §3.b.iv.

"Contribution distributed in accordance with [...] (the EPL)" while the term 'Contribution' refers – besides other elements – to "changes to the Program, and additions to the Program"[94]. Unfortunately, this is a circular definition: 'Program' is defined by 'Contribution'; and 'Contribution' is defined by 'Program'. Nevertheless, one has to read the license benevolently. Uncontroversial should be this: If one distributes any modified EPL licensed program, library, module, or plugin, then one has to publish the modified source code, too. If one "adds" some own plugins or additional libraries which are used by an EPL licensed program (which on behalf of this use must have been modified by adding [sic!] procedure calls) then one has to publish the code of both parts: that of the program and that of the added elements. In this sense, the EPL clearly protects the binaries against re-closings like other weak copyleft using licenses. But if one distributes only an EPL licensed library which is used as a component by another not EPL licensed on-top program, then this library does not depend on the top development – provided that the library itself does not call any (program) functions or procedures delivered by the overarching on-top development. Hence, nothing is added to the library; and hence, no other code than that of the library must be published. Therefore, the EPL does not use the strong copyleft effect in the meaning of – for example – the GPL.

## 2.6 The protecting power of the European Union Public License (EUPL)

As an approved *open source license*[95], the European Union Public License[96] protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[97]. Furthermore, based on its patent clause[98], the EUPL protects the users against patent disputes[99]. Besides

[94] cf. *Open Source Initiative*: EPL-1.0, 2005, wp. §1.

[95] cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

[96] The European Union Public License, version 1.1 is maintained by the European Union and hosted under the label "Joinup" (cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp). This EUPL has officially been translated into many languages, among others into German (cf. *Europäische Gemeinschaft a. European commission Joinup*: EUPL-1.1/DE, 2007, wp). Because of this multi lingual instances, the OSI does not offer its own version, but just a landing page linked to the lading page of the European host "Joinup" (cf. *Open Source Initiative*: European Union Public License, version 1.1 (EUPL-1.1; 2007 [n.y. of the html page itself] ⟨URL: `http://opensource.org/licenses/EUPL-1.1`⟩ – reference download: 2013-03-04, wp).

[97] cf. *Europäische Gemeinschaft a. European commission Joinup*: EUPL-1.1/DE, 2007, wp. §2.

[98] → OSLiC pp. 53

[99] cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp. §2, at its tail.

this patent clause, the EUPL additionally contains a "Disclaimer of Warranty" and a "Disclaimer of Liability"[100]. These three elements together protect the contributors / distributors against patents disputes and warranty claims. Finally, the EUPL also protects the distributed sources against a re-closing / re-privatizing and the contributors against the loss of feedback. This protection is based on two steps: First, the European Public License contains a particular paragraph titled "Copyleft clause" which stipulates that "copies of the Original Work or Derivative Works based upon the Original Work" must be distributed "under the terms of (the European Union Public) License"[101]. Second, the EUPL requires that each licensee – as long as he "[...] continues to distribute and/or communicate the Work" – has also to "[...] provide [...] the Source Code", either directly or by "[...] (indicating) a repository where this Source will be easily and freely available [...]"[102]. This condition ssems to be so important for the EUPL that the license repeats its message: in another paragraph the EUPL requires again that "if the Work is provided as Executable Code, the Licensor provides in addition a machine-readable copy of the Source Code of the Work along with each copy of the Work [...] or indicates, in a notice [...], a repository where the Source Code is easily and freely accessible for as long as the Licensor continues to distribute [...] the Work"[103]. Based on the meaning of "Work" which is defined by the EUPL as "the Original Work and/or its Derivative Works"[104] it must be concluded that the EUPL is a copyleft license.

But nevertheless, the European Union Public License is not a license with strong copyleft: On the one hand, if one takes the core of the EUPL then the license seems to protect not only the modifications of the original work against re-closings and (re-)privatizings, but also the on-top developments because normally you have to publish the source code in both cases. Understood in this way, the EUPL would be a 'strong copyleft license'. But on the other hand, the EUPL additionally contains a "Compatibility clause" stating that "if the Licensee Distributes [...] Derivative Works or copies thereof based upon both the Original Work and another work licensed under a Compatible Licence, this Distribution [...] can be done under the terms of this Compatible Licence"[105] – while the term "Compatible Licence" is explicitly defined by a list of compatible licenses, for example the Eclipse Public License[106]. Based on this compatibility clause the obligation to publish the code of an on-top development can be subverted: As first step, you could release a little, more or less futile on-top application licensed under the

[100] cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp. §7 & §8.
[101] cf. id., l.c., wp. §5.
[102] cf. id., ibid.
[103] cf. id., l.c., wp. §3.
[104] cf. id., l.c., wp. §1.
[105] cf. id., l.c., wp. §5.
[106] cf. id., l.c., wp. Appendix.

Eclipse Public License[107] which uses a library licensed under the EUPL. As second step, you add this 'EUPL library' which you now may also distribute under the EPL instead of retaining the EUPL licensing. So, finally you obtain the same work under the Eclipse Public License which is a weak copyleft license[108]. Hence the protection of the EUPL-1.1 is not as comprehensive as one might assume on the base of the license text itself[109], it can at most be a weak copyleft license – even if the reader might get the impression that the authors of the EUPL wished to write a strong copyleft license. Howsoever, the EUPL license does not protect the on-top developments against a privatizing.

## 2.7 The protecting power of the GNU General Public License (GPL)

The GNU General Public License – also known as GPL – is maintained and offered by the Free Software Foundation and hosted as part of the well known "GNU operating system homepage"[110]. Currently, there are two versions of the GPL which are classified as OSI approved open source licenses[111], the GPL-2.0[112] and the GPL-3.0[113]. Although both versions of the GPL are aiming for the same results and the same spirit, they differ with respect to textual and arguing

[107] Taking the license text very seriously, it is not even necessary that this little futile application must depend on the EUPL library by calling functions of EUPL library. The license text only says that "another [any other] work licensed under a Compatible Licence" can be distributed together with "derivative works". By this wording, the license itself is establishing a contrast between the derivative work and the other work – what indicates that the other work has not necessarily also to be a derivative work.

[108] → OSLiC, p. 30

[109] This kind of specifiying the protective power of the EUPL is initially presented by the FSF (cf. *Free Software Foundation*: Various Licenses and Comments about Them, 2013, pp. wp. section 'European Union Public License'). The EU answers that publishing such a trick will comprise its user in the eyes of the open source community (cf. *European Community* a. *European commission Joinup*: New FSF statements on the EUPL are a step in the right direction; 2013 [n.y] ⟨URL: `https://joinup.ec.europa.eu/community/eupl/news/new-fsf-statements-eupl-are-step-right-direction`⟩ – reference download: 2013-03-05, p. wp). That is undoubtely true. But unfortunately, this argument does not close the hole in the protecting shield put up by the EUPL.

[110] cf. *Free Software Foundation*: GNU Operating System[:] Licenses; 2011 ⟨URL: `http://www.gnu.org/licenses/`⟩ – reference download: 2013-03-25, wp.

[111] cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

[112] For the original version, offered by the FSF cf. *Free Software Foundation*: The GPL-2.0 License (FSF), 1991, wp. For the version, offered by the OSI cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp.

[113] For the original version, offered by the FSF cf. *Free Software Foundation*: GNU General Public License [version 3]; 2007 [n.y. of the html page itself] ⟨URL: `http://www.gnu.org/licenses/gpl.html`⟩ – reference download: 2013-03-06, wp. For the version, offered by the OSI cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp.

structure. Therefore, it should be helpful to treat these two licenses separately.

#### 2.7.0.1 GPL-2.0

The protecting power of the GPL-2.0 can easily be determined: First, the license allows the users of a received software to "copy and distribute" unmodified "copies of the [...] source code"[114] as well as to "[...] modify [...] copies [...] or any portion of it, [...] and (to) distribute such modifications [...]"[115] – not only in the form of source code, but also in the form of binaries[116]. Thus – and in accordance of being an approved *open source license*[117] – the GPL-2.0 protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries. Second, it protects the contributors against warranty claims[118] and – based on its copyleft effect[119] – also against the loss of feedback. Third, the GPL-2.0 protects the source code itself in a nearly complete mode against privatizings: even if one initially distributes only the binary version of a modification which one has generated (as a "work based on the" original) by "copying" any portion of the original work into this new derivative work[120], then one has nevertheless to offer a possibility to get the source code[121] – namely for "the modified work as whole"[122]. This modified "work based on the [original] Program" has to be read in a very broad sense; it "[...] means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language"[123]. Hence, in the context of software distribution, the GPL-2.0 does not only protect the software against re-privatizings, but also possible on-top developments against privatizings.

But the GPL-2.0 does not protect against patent disputes[124] – neither the users, nor the contributors or distributors – and it does not protect the (modified) software which is not distributed against (re-)privatizings[125].

---

114) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §1.
115) cf. id., l.c., wp. §2.
116) cf. id., l.c., wp. §3.
117) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.
118) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §11, §12.
119) cf. id., l.c., wp. §3.
120) cf. id., l.c., wp. §2.
121) cf. id., l.c., wp. §4.
122) cf. id., l.c., wp. §3.
123) cf. id., l.c., wp. §0.
124) → OSLiC, p. 53
125) This is a 'lack' in the GPL which the AGPL wants to close: you are indeed allowed to modify and install a GPL-2.0 licensed server software own your own machine for offering a service based on this modified software without being obliged to give your improvements back to the community. But – at least in Germany – this viewpoint seems to have to respect

#### 2.7.0.2 GPL-3.0

An important modification of the GPL-3.0 is evoked by the use of the new wording to "propagate" or to "convey" a "covered work": On the one hand a "covered work" denotes "either the unmodified Program or a work based on the Program". This "work based on the Program" is defined as a "modified version" of an "earlier" instance of the program which has been derived from this earlier instance by "(copying it) from or (adapting) all or part of it" in way other than exactly copying the earlier instance[126]. On the other hand, "to propagate a work" denotes "copying, distribution (with or without modification), making available to the public" and any other kind of treating the work "[...] except executing it on a computer or modifying a private copy"[127]. Third, the GPL 3.0 specifies that to "convey" a work "[...] means any kind of propagation that enables other parties to make or receive copies"[128]. This specification shall later on help to clarify that it is an act of distribution if the recipient himself actively copies or fetches a program.

Referring to this new wording, the GPL-3.0 allows – as a "basic permission" – to "[...] make, run and propagate covered works [...] without conditions so long as your license otherwise remains in force"[129]. This might be read as *anything is allowed without any restrictions – provided there does not exist any rule which must be respected*. Based on these specifications, the use and the modification of a GPL-3.0 program only for yourself is not restricted[130].

So, in general – like all the other open source licenses and in accordance to the

rigorous limits. Sometimes, it is said that even distributing software over the compartments of a holding is already a distribution which – in the case of a GPL-2.0 licensed software – would evoke the obligation to distribute the source code, too. [IMPORTANT: citation still needed!!!]

126) cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §0.

127) cf. id., ibid.. The GPL 3.0 wants to cover the copyright systems of all countries of the world without dealing with their particular constraints directly. Therefore it generally states, that the meaning of the phrase "to propagate a work" – in the spirit of the FSF – is whatever the specific copyright system wants to be covered by these words, "[...] except executing it on a computer or modifying a private copy".

128) cf. id., ibid.

129) cf. id., l.c., wp. §2.

130) In general, you have to infer that you do not have to fulfill any tasks if you are using a piece of open source software only for yourself – namely based of the fact that the particular license rules focus only on the distribution of the software, not on the private use. But in the GPL-3.0, this assertion concerning the private use becomes more explicit: It is one of your "basic permissions" to "[...] make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force". And "to propagate a work" refers to anything "[...] except executing it on a computer or modifying a private copy" (cf. id., l.c., wp. §2 and §0). Thus, the GPL-3.0 supports your total freedom on your own machine: Do whatever you want to do; anything goes – as long as you do not hand the result over to any third party in any sense.

OSD[131] – also the GPL protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[132]. Furthermore, based on its patent clauses, the GPL-3.0 protects the users and the contributors of a software against patent disputes[133]. Additionally, the GPL-3.0 tries to protect the contributors or distributors against warranty claims by its well known "Disclaimer of Warranty"[134] and "Limitation of Liability"[135] which must explicitly made been known at least in each case of source code distribution[136]. Finally, the most forceful protection of the GPL-3.0 concerns the protection against the loss of feedback and against the privatization: Whenever you distribute a GPL-3.0 licensed program in the form of binaries, you have to make the source accessible, too[137]. Moreover, this obligation concerns every covered work, hence not only the unmodified original, but also any modification or adaption derived by any other kind of copying parts of the original into the "resulting work"[138]: "You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License"[139]. So, no doubt: the GPL wants also the source code of all on-top developments to be published, not only the modified programs and libraries used as base of these on-top developments. The single mode of use, the GPL does not protect against privatizings, is the mode of using the software only for yourself[140].

## 2.8 The protecting power of the GNU Lesser General Public License (LGPL)

The LGPL is maintained and offered by the Free Software Foundation and hosted as part of the well known "GNU operating system homepage"[141]. The meaning of the name *LGPL* was changed in the course of time. First, in 1991, it should be resolved as "GNU Library General Public License" and should denote the "first released version of the library GPL" which was "[...] numbered 2 because it goes with version 2 of the ordinary GPL". Today, this license is marked as "superseded

131) cf. *Open Source Initiative*: The Open Source Definition, 2012, wp.
132) cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §3, §4, §5, and §6.
133) → OSLiC, p. 54
134) cf. id., l.c., wp. §15.
135) cf. id., l.c., wp. §16.
136) cf. id., l.c., wp. §4.
137) cf. id., l.c., wp. §6.
138) cf. id., l.c., wp. §0.
139) cf. id., l.c., wp. §6.
140) Quite the contrary: The GPL-3.0 explicitly allows to delegate the modification to third parties and allows to distribute the source code as working base "[...] to others for the sole purpose of having them make modifications exclusively for you [...]" (cf. id., l.c., wp . §2).
141) cf. *Free Software Foundation*: The GNU OS Licenses, 2011, wp.

by the GNU Lesser General Public License"[142]. This newer *LGPL* version from 1999 was released as "the successor of the GNU Library Public License, version 2, hence [as] the version number 2.1"[143]. Finally, in June 2007, the – for now – last version of the *LGPL* was released – namely with a new structure: While GPL-2.0 and LGPL-2.1 are similar, but independent licenses, the LGPL-3.0 has to be read as an addendum to GPL-3.0: At the beginning of the LGPL-3.0 license, the content of the corresponding GPL-3.0 was included into the LGPL by the sentence that "this version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below"[144]. Based on these differences, it seems to be suitable to treat the different LGPLs separately.

#### 2.8.0.3 LGPL-2.1

Like the other versions of the GPL or LPGL, the LGPL-2.1 also explicitly describes its purpose as the task to "protect" the "rights" of the software users: it states that generally all "[...] the GNU General Public Licenses are intended to guarantee your freedom to share and change free software [...]"[145]. Thus – of course – the LGPL-2.1 is an approved *open source license*[146] which protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[147]. But the LGPL-2.1 does not offer any sentences to infer that it grants any patent rights to the software user[148]. So, it does not protect anyone against patent disputes, neither the users, nor the contributors / distributors. Instead of this, the LGPL-2.1 contains a special section "No Warranty" offering two paragraphs which together establish the protection of the contributors and distributors against warranty claims[149]. Finally, the LGPL-2.1 also protects the distributed sources against a re-closing / re-privatizing and the contributors against the loss of feedback. For that purpose, the LGPL-2.1 on the one hand states that the recipient "[...] may modify (his) copy or copies of the Library or any portion of it [...] and copy and distribute such modifications

142) cf. *Free Software Foundation*: GNU Library General Public License [version 2.0]; 1991 [n.y. of the html page itself] ⟨URL: `http://www.gnu.org/licenses/old-licenses/lgpl-2.0.html`⟩ – reference download: 2013-03-25, wp.

143) cf. *Free Software Foundation*: GNU Lesser General Public License [Version 2.1]; 1999 [n.y. of the html page itself] ⟨URL: `http://www.gnu.org/licenses/lgpl-2.1.html`⟩ – reference download: 2013-03-06, wp.

144) cf. *Free Software Foundation*: GNU Lesser General Public License [version 3]; 2007 [n.y. of the html page itself] ⟨URL: `http://www.gnu.org/copyleft/lesser.html`⟩ – reference download: 2013-03-06, wp.

145) cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. Preamble.

146) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

147) cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §1, §2, §4.

148) → OSLiC, p. 55

149) cf. id., l.c., wp. §15, §16.

[...]" provided that the results of these modifications are "[...] licensed at no charge to all third parties under the terms of (the LGPL-2.1)"[150]. On the other hand, this LGPL version allows to distribute such modifications "in object code or executable form" provided that one accompanies these entities "[...] with the complete corresponding machine-readable source code" which itself must be distributed under the terms of the LGPL-2.1[151].

But in opposite to the GPL, the LGPL does not require to publish the code of an overarching program or any on-top development: It distinguishes the "work that *uses* the Library" from the "work *based on* the Library": First, it defines the "Library" as any "software library or work" licensed under the LGPL-2.1 and adds that "a 'work *based on* the Library' means either the Library or any derivative work under copyright law"[152]. Second, it defines the "work that *uses* the Library" as any "[...] program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it" whereas this "work that *uses* the Library" – taken "in isolation" – clearly "[...] is not a derivative work of the Library [...]"[153]. Third – and explictily "as an exception to the Sections above" – the LGPL-2.1 allows to "[...] combine or link a 'work that uses the Library' with the Library to produce a work containing portions of the Library, and distribute that work under terms of (one's own) choice" provided one "(accompanies) the work with the complete corresponding machine-readable source code for the Library". Together, these three specifications clearly require that one must publish / distribute the source code of the library itself – regardless, whether it is modified or not, and regardless, whether one distributes the code directly or makes 'only' written offer for receiving the source code of the library separately[154]. But these specifications do not require that one also must publish / distribute the source code of the *work that uses the library* or – as the OSLiC is using to say – the *the on-top developments*.

Thus – no surprise – it has to be inferred that the LGPL does not protect the on-top developments against a privatizing. And of course, that is the reason why it is called the *GNU* Lesser *General Public License*.

[150] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §2.
[151] cf. id., l.c., wp. §4.
[152] cf. id., l.c., wp. §0, emphasis ours.
[153] cf. id., l.c., wp. §5, emphasis ours. To be exact: the LGPL states also, that this work can nevertheless become a derivative work under the particular circumstances of being linked to the library. But even then, the LGPL allows to treat this 'derivative work' as a work which is not a derivative work, provided one fulfills some additional conditions. With respect to this viewpoint, the hint of the LGPL that the non-derivative work becomes a derivate work by linking it, seems not to be as crucial as one might expect.
[154] cf. id., l.c., wp. §6.

#### 2.8.0.4 LGPL-3.0

The LGPL-3.0 wants to be read as an extension of the GPL-3.0. For that purpose, it explicitly "[...] incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by (some) additional permissions [...]"[155]. Thus, the LGPL-3.0 inherits the most parts of the protecting power of the GPL-3.0 – except those parts which deal with the overarching on-top development: In opposite of the GPL-3.0, the LGPL allows to embed LGPL-3.0 licensed libraries into libraries of higher complexity[156], into on-top applications[157] and into sets of reorganized library systems[158]. Moreover, the LGPL-3.0 allows to "convey" these overarching units "under terms of (one's own) choice"[159]. Therefore, one is not necessarily obliged to publish the source code of these on-top developments, too[160] – but, of course, one is obliged to publish the source code of the (modified) embedded libraries themselves.

Based on the already described protecting power of the GPL-3.0[161] and on these additional specifications of the LGPL-3.0, one can summarize the protecting power of the LGPL-3.0 this way:

First, the LGPL protects the users against the loss of the right to use, to modify and/or to distribute the received software. Additionally, it protects them against patent disputes. Second, it protects the contributors and distributors against the loss of feedback, against warranty claims and against patent disputes. Finally, it protects the distributed software itself against re-privatizings.

But the LGPL-3.0 does not protect the undistributed source code and does not protect the on-top developments against privatizings.

---

155) cf. *Free Software Foundation*: The LGPL-3.0 License (FSF), 2007, wp., just before §0..
156) cf. id., l.c., wp. §3.
157) cf. id., l.c., wp. §4.
158) cf. id., l.c., wp. §5.
159) cf. id., l.c., wp. §4.
160) To be exact: The LGPL-3.0 wants to assure that "combined works" can be re-combined on the base of newer versions of the embedded library. For that purpose, one has either to use "a suitable shared libary mechanism" which allows to replace the embedded library without relinking the larger unit, or one has to publish at least "the minimal corresponding source [code]" and a set of binaries by which the user himself can relink the overarching unit on the base of a newer version ob the embedded library (cf. id., ibid.)
161) → OSLiC, p. 35

## 2.9 The protecting power of the MIT license

As an approved *open source license*[162], the MIT License[163] protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[164]. Additionally, it protects the contributors and/or distributors against warranty claims of the software users, because it contains a 'No Warranty Clause'[165]. And finally it protects the distributed sources against a change of the license which would close the sources, because the "permission [...] to use, copy, modify, [...] distribute, [...] (is granted) subject to the [...] conditions, [that] the [...] copyright notice and this permission notice shall be included in all copies or substantial portions of the Software"[166].

But the MIT License does not protect the users or the contributors and/or distributors against patent disputes (because it does not contain any patent clause). Additionally, it does not protect the contributors against the loss of feedback (because it does not 'copyleft' the software). Moreover, the MIT license does not protect the undistributed software or the distributed binaries against re-closings – neither in unmodified nor in modified form – because it allows to redistribute only the binaries without also supplying the source code[167]. Finally, the MIT license does not protect the on-top developments against a privatizing.

[162] cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

[163] 'MIT' has to be resolved as "Massachusetts Institute of Technology" (cite[(cf.][wp]wpMitLic2011a).

[164] cf. *Open Source Initiative*: The Open Source Definition, 2012, wp 1ff.

[165] cf. *Open Source Initiative*: The MIT License; 2012 [n.y.] ⟨URL: `http://opensource.org/licenses/mit-license.php`⟩ – reference download: 2012-08-24, wp.

[166] cf. id., ibid.. The argumentation why the source code is protected, but not the binary form follows that of the BSD licenses: By these requirements, one is not obliged to redistribute the sourcecode of a modified (derivative) work. One is allowed to modify a received version and to distribute the results only in binary form and to keep one's improvements closed. But if one distribute the source code of the modifications, the licensing is retained, simply because the MIT "[...] permission note shall be included in all copies or substantial portions of the software".

[167] cf. id., ibid.

## 2.10 The protecting power of the Mozilla Public License (MPL)

As an approved *open source license*[168], the Mozilla Public License[169] protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[170]. Furthermore, based on its split and distributed patent clause[171], the MPL protects the users against patent disputes[172]. Besides this patent sections, the MPL additionally contains a "Disclaimer of Warranty" and a "Limitation of Liability"[173]. These three elements together protect the contributors / distributors against patents disputes and warranty claims. Finally, the MPL also protects the distributed sources against a re-closing / re-privatizing and the contributors against the loss of feedback: The MPL clearly says that, on the one hand, "all distribution of Covered Software in Source Code Form, including any Modifications [...] must be under the terms of this License"[174] and that, on the other hand, an MPL licensed software "[...] (distributed) in Executable Form [...] must also be made available in Source Code Form [...]"[175]. So, it must be inferred that the MPL is a copyleft license.

But nevertheless, the Mozilla Public License is not a license with strong copyleft. It does not protect on-top developments against privatizings: First, the MPL does

168) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

169) In 2012, the Mozilla Public License 2.0 (cf. *Mozilla Foundation*: Mozilla Public License 2.0 (MPL-2.0); 2012 ⟨URL: `http://www.mozilla.org/MPL/2.0/`⟩ – reference download: 2013-03-05, wp) has been released as a result of a longer "Revision Process"(cf. *Mozilla Foundation*: About MPL 2.0: Revision Process and Changes FAQ; 2013 [n.y.] ⟨URL: `http://www.mozilla.org/MPL/1.1/`⟩ – reference download: 2013-03-05, wp) by which the Mozilla Public License 1.1 (cf. *Mozilla Foundation*: Mozilla Public License Version 1.1; 2013 [n.y.] ⟨URL: `http://www.mozilla.org/MPL/1.1/`⟩ – reference download: 2013-03-05, wp) has been ousted. The OSI is also hosting its version of the MPL-2.0 (cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp) and is listing it as an OSI approved license (cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp) while it classifies the MPL-1.1 as a "superseded license"(cf. *Open Source Initiative*: The Open Source Licenses by Category, 2013, wp). The Mozilla Foundation itself says concerning the difference between the two licenses that "the most important part of the license – the file-level copyleft – is essentially the same in MPL 2.0 and MPL 1.1" (cf. *Mozilla Foundation*: MPL 2.0: Revision Process and Changes, 2013, wp). By reading the MPL-1.1, one could get the impression that fulfilling all conditions of the MPL-2.0 would imply also to act in accordance to the MPL-1.1. Thus the OSLiC focuses on the MPL-2.0, at least for the moment. Nevertheless, in this section we want to use the general label 'MPL' without any releasenumber for indicating that with respect to its protecting power the MPL-2.0 and the MPL-1.1 can be taken as equipollent.

170) cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §2.1.a.

171) → OSLiC pp. 56

172) cf. id., l.c., wp. §2.1.b, §2.3, §5.2.

173) cf. id., l.c., wp. §6 & §7.

174) cf. id., l.c., wp. §3.1.

175) cf. id., l.c., wp. §3.2.

not use the term *derivative work*[176]. Instead of this, the MPL denotes the "[...] (initial) Source Code Form [...] and Modifications of such Source Code Form" by the label "Covered Software"[177] – while the term "Modifications" refers to "any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software or any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software"[178]. Second, the MPL contrasts the source code form and its modifications with the "Larger Work" by specifying that the larger work is "[...] material, in a seperate file or files, that is not covered software"[179]. Finally, the MPL states, that "you may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software"[180]. Based on these specifications, one has to reason that an on-top development which depends on MPL licensed libraries by calling some of their functions, is undoubtably a derivative work[181], but also only a larger work in the meaning of the MPL so that code of this on-top application needs not to be published – provided, that the library and the on-top development are distributed as different files[182]. Hence, the MPL is license with a weak copyleft effect and does not protect the on-top developments against privatizings.

[176] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. The MPL-1.1 uses the term *derivative work* only in the context of writing new "versions of the license", not in the context of licensing software (cf. *Mozilla Foundation*: Mozilla Public License Version 1.1, 2013, wp. §6.3).

[177] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §1.4.

[178] cf. id., l.c., wp. §1.10. The Mozilla Foundation denotes this reading by the term "file-level copyleft" (cf. *Mozilla Foundation*: MPL 2.0: Revision Process and Changes, 2013, wp).

[179] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §1.7.

[180] cf. id., l.c., wp. §3.3.

[181] This follows from the general meaning of a *derivative work* as a benevolent software developer would read this term (→ OSLiC, pp. 63). But again: The MPL does not focus on this general aspect; it uses its own concept of a *larger work*.

[182] It might be discussed whether integrating a declaration of a function, class, or method into the on-top development by including the corresponding header files indeed means that one is "including portions (of the Source Code Form)" into a file which therefore has to be taken as "Modification" (cf. *Mozilla Foundation*: Mozilla Public License Version 1.1, 2013, wp. §1.4). From the viewpoint of a benevolent developer it should be difficult to argue that the including of declaring (header) files alone can evoke a derivative work. It is the call of the function in one's code which establishes the dependency. But that is not the point, the MPL focuses. The MPL aims on the textual reuse of (defining) code snippets. Hence, one could ignore the textual integration of parts of the declaring header files: it should not trigger that one's own work becomes a modification in the eyes of the Mozilla Findation. But of course, one would circumvent the idea of the MPL if one hides defining code in header files and reuses that code by one's own compilation. This would undoubtably be an incorporation of portions and therefore would make the incorporating file becoming a modification of the MPL licensed initial work.

## 2.11 The protecting power of the Microsoft Public License (MS-PL)

As an approved *open source license*[183], the Microsoft Public License protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[184]. Furthermore, based on its patent clause[185], the MS-PL protects the users against patent disputes[186]. Because of this patent clause and of its concise *disclaimer of warranty*, the MS-PL also protects the contributors / distributors against patents disputes and warranty claims[187]. Finally, the Microsoft Public License protects the distributed sources themselves – and even "portions of these sources" – *against* a change of the license which would *reset* the work *as closed software*, because first, one "[...] must retain all copyright, patent, trademark, and attribution notices that are part of the software"[188], and because second, one must also incorporate "a complete copy of this license" into one's own distribution premised one distributes the source code[189].

But the Microsoft Public License does not protect the contributors against the loss of feedback because it does not 'copyleft' the software: The license does not contain any sentence which requires that one has to publish the sources, too[190]. In the same spirit, the MS-PL does not protect the undistributed software or the distributed binaries against re-closings – neither in unmodified nor in modified form – because the MS-PL License allows to (re)distribute the binaries without also supplying the sources – even if the binaries rest upon sources modified by the

183) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.

184) cf. *Open Source Initiative*: Microsoft Public License (MS-PL); 2013 [n.y.] ⟨URL: `http://opensource.org/licenses/MS-PL`⟩ – reference download: 2013-02-26, wp. §2.

185) → OSLiC pp. 57

186) cf. id., l.c., wp. §2.B and §3.B.

187) cf. id., l.c., wp. §2B, §3B, §3D.

188) cf. id., l.c., wp. §3C.

189) cf. id., l.c., wp. §3D.

190) There seems to be some misunderstandings on the internet: The English wikipedia specifies the MS-PL as a permissive license and the MS-RL as a license with copyleft effect (cf. *Wikipedia (en)*: Shared source; n.l, 2013 [n.y.] ⟨URL: `http://en.wikipedia.org/wiki/Shared_source`⟩ – reference download: 2013-02-26, wp). The German wikipedia says that the MS-PL is a license with a "schwachen [weak] copyleft" (cf. *Wikipedia (de)*: Microsoft Public License; n.l, 2013 [n.y.] ⟨URL: `http://de.wikipedia.org/wiki/Microsoft_Public_License`⟩ – reference download: 2013-02-26, wp). And it says also that the "Microsoft Reciprocal License" (MS-RL) is a license with weak copyleft, too (cf. *Wikipedia (de)*: Microsoft Reciprocal License; n.l, 2013 [n.y.] ⟨URL: `http://de.wikipedia.org/wiki/Ms-RL`⟩ – reference download: 2013-02-26, wp). But for the very thoroughly working "ifross license center", the MS-RL is a license with restricted (weak) copyleft, while the MS-PL is a permissive license with some selectable options (cf. *ifross*: ifross Lizenz-Center, 2011, wp). Based on the license text itself and these other readings, we decided to take the MS-PL as a permissive license in accordance to the English wikipedia page and the ifross page.

distributor. Finally, also the MS-PL does not protect the on-top developments against a privatizing.

## 2.12 The protecting power of the Postgres License (PgL)

As an approved *open source license*[191], the PostgreSQL License protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[192]. Because of its *disclaimer of warranty*, the PgL also protects the contributors / distributors against warranty claims[193]. Finally, the PgL protects the distributed sources themselves *against* a change of the license which would *reset* the work *as closed software*, because the "copyright notice" and the whole license must "[...] appear in all copies"[194].

But the PostgreSQL License does not protect the contributors against the loss of feedback because it does not 'copyleft' the software: The license does not contain any sentence which requires that one has to publish the sources, too. In the same spirit, the PgL does not protect the undistributed software or the distributed binaries against re-closings – neither in unmodified nor in modified form – because the PgL allows to (re)distribute the binaries without also supplying the sources – even if the binaries rest upon sources modified by the distributor. Finally, the PgL does not protect the on-top developments against a privatizing.

## 2.13 The protecting power of the PHP License

As an approved *open source license*[195], the PHP-3.0 License protects the user against the loss of the right to use, to modify and/or to distribute the received copy of the source code or the binaries[196]. Because of its *disclaimer of warranty*, the PHP license also protects the contributors / distributors against warranty claims[197]. Finally, the PHP license protects the distributed sources themselves *against* a change of the license which would *reset* the work *as closed software*, because "redistributions of source code must retain the [...] copyright notice, this list of conditions and the [...] disclaimer"[198].

191) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.
192) cf. *Open Source Initiative*: The PostgreSQL Licence (PostgreSQL); 2013 [n.y.] ⟨URL: `http://opensource.org/licenses/PostgreSQL`⟩ – reference download: 2013-02-27, wp.
193) cf. id., ibid.
194) cf. id., ibid.
195) cf. *Open Source Initiative*: The Open Source Licenses, alphabetically sorted, 2012, wp.
196) cf. *Open Source Initiative*: The PHP License 3.0 (PHP-3.0); 2013 [n.y.] ⟨URL: `http://opensource.org/licenses/PHP-3.0`⟩ – reference download: 2013-02-27, wp.
197) cf. id., ibid.
198) cf. id., ibid.

But the PHP-3.0 License does not protect the contributors against the loss of feedback because it does not 'copyleft' the software: The license does not contain any sentence which requires that one has to publish the sources, too. In the same spirit, the PHP license does not protect the undistributed software or the distributed binaries against re-closings – neither in unmodified nor in modified form – because the PHP license allows to (re)distribute the binaries without also supplying the sources – even if the binaries rest upon sources modified by the distributor.

## 2.14 Summary

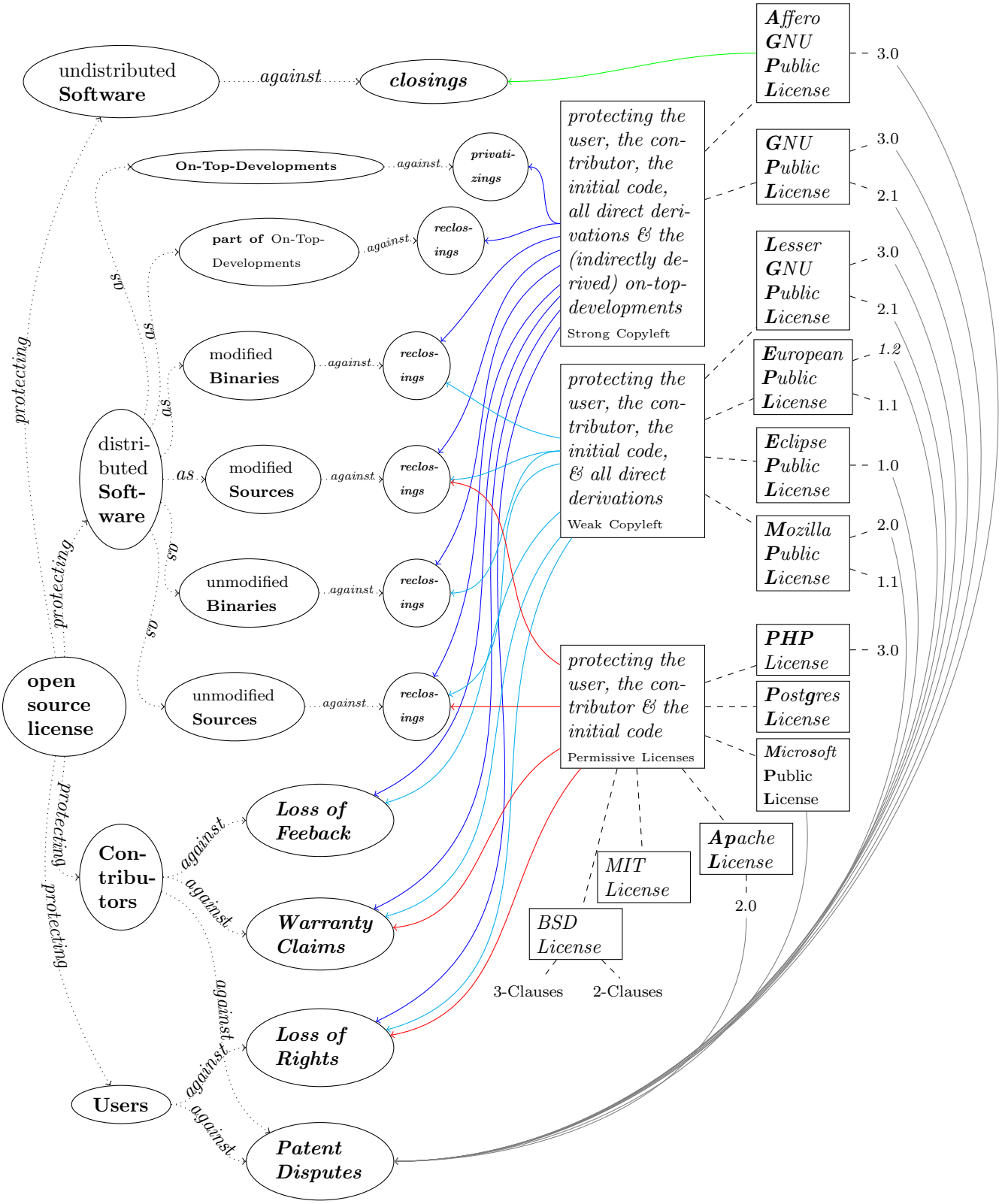
All these specifications can not only be summarized by a table[199], but also by a mindmap as it is shown at the end of this chapter. Moreover, based on these specifications, one could generate new groups of open source licenses, new classes, like 'user protecting licenses'[200], 'patent disputes fending licenses' up to more sophisticated taxonomies.

However, one must keep in mind that all of these grouping viewpoints do not legitimate the conclusion that all members of a group can be respected by fulfilling the same requirements. This would only be possible if the grouping criteria would directly refer to the fulfilling tasks. Indeed, nearly all open source licenses do differ with respect to these criteria, and even if the differences are very small, they can't be neglected[201]. So: reflecting on possible classes of open source licenses is a good method to become familiar with the area of open source licenses. But it is not a method to determine, what needs to be done to obtain the right to use the software. For that purpose every license must be considered individually.

[199] → OSLiC, p. 27

[200] all of them because all of them have to fulfill the OSD

[201] Pars pro toto: Both, the BSD license and the Apache license require that you provide an indication to the developers of the application. But in case of the BSD license you have to publish the copyright notice / line, while in case of the Apache license you have exactly to present the content of the notice file distributed together with the application.


undistributed Software
against
closings
On-Top-Developments
against
privatizings
part of On-Top-Developments
against
reclosings
modified Binaries
against
reclosings
modified Sources
against
reclosings
unmodified Binaries
against
reclosings
unmodified Sources
against
reclosings
distributed Software
as
as
as
as
as
protecting
protecting
open source license
protecting
protecting
protecting
Contributors
Users
against
against
against
against
against
Loss of Feeback
Warranty Claims
Loss of Rights
Patent Disputes
protecting the user, the contributor, the initial code, all direct derivations & the (indirectly derived) on-top-developments
Strong Copyleft
protecting the user, the contributor, the initial code, & all direct derivations
Weak Copyleft
protecting the user, the contributor & the initial code
Permissive Licenses
Affero GNU Public License
GNU Public License
Lesser GNU Public License
European Public License
Eclipse Public License
Mozilla Public License
PHP License
Postgres License
Microsoft Public License
Apache License
MIT License
BSD License
3-Clauses
2-Clauses
3.0
3.0
2.1
3.0
2.1
1.2
1.1
1.0
2.0
1.1
3.0
2.0

# 3 Open Source: About Some Side Effects

## 3.1 The problem of implicitly releasing patents

> *In this chapter, we are briefly analyzing the effect of patent clauses in open source licenses – not in general, but with respect to the license fulfilling tasks they require, also known as the 'implicit acceptance of a patent use' by distributing open source software.*

At least the free software movement frowns on the existence of software patents[202]. One of the most known witnesses for that attitude is the GPL itself. Its preamble purports that "[...] any free program is threatened constantly by software patents"[203]. One can read that the open source community fears three risks: First, they are apprehensive of people who hijack the idea of a piece of open source software they do not have developed, register a corresponding patent, and finally try to earn money by preventing the use of the software or by envolving its users into patent ligitations[204]. Second, they fear a bramble of general software patents which practically prohibits them to develop open source software legally[205]. Third, they anticipate the possibility that (not quite benevolent) open

[202] For an early and elaborated description on the effects of software patents based on the viewpoint of the free software movement see *Stallman, Richard M.*: Free Software: Freedom and Cooperation; transcript of a speech given at New York University on 29 May 2001; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, wp. This lecture seems to have been given more than once and printed later on (cf. *Stallman, Richard M.*: The Danger of Software Patents; transcript of a speech given at University of Cambridge, London on the 25th of March 2002; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, wp). Within the first decade of 2000, the focus switched to a more political fight against software patents (cf. *Stallman, Richard M.*: Fighting Software Patents - Singly and Together; n.st. [2004] ⟨URL: `http://www.gnu.org/philosophy/fighting-software-patents.html`⟩ – reference download: 2013-02-18, wp). But recently there seems to have been appeared another turn in dealing with software patents: Not fighting against the patents, but mitigating their effects: The proposal is '[...] (to legislate) that developing, distributing, or running a program on generally used computing hardware does not constitute patent infringement' (cf. *Stallman, Richard M.*: Let's Limit the Effect of Software Patents, Since We Can't Eliminate Them; in: Wired, n.st. January (2012) ⟨URL: `http://www.wired.com/opinion/2012/11/richard-stallman-software-patents/`⟩ – reference download: 2013-02-18, ISSN n.st., wp)

[203] cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, p. wp.

[204] cf. *Jaeger a. Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software, 2011, p. 234.

[205] cf. id., ibid.

source developers could try to register patents for undermining the open source principles[206].

Howsoever, regardless whether one tries to fight against software patents or not, software patents have come true. To act law-abidingly requires to manage the constraints of patents properly. Open source licenses know and respect this necessity. Moreover, at least some of them try to manage the effect of software patents by specific patent clauses[207] or by several sentences distributed in the license text[208]. But why does the OSLiC have to deal with this topic, if the OSLiC does not want to participate in general discussions?

In opposite to the other conditions of the open source licenses, their patent clauses or propositions in general do not directly refer to a specific set of actions which has to be executed for acting in accordance to the licenses. Open source patent clauses normally do not join in the game 'paying by doing'. So, actually, it does not seem to be necessary to mention the patent clauses here.

Unfortunately, although the patent clauses do not directly say *'do this or that in these or those circumstances'*, some of them nevertheless trigger side effects which evoke that the distributors of open source software implicitly having already something done if they actually are distributing a piece of open source software. This implicit effect makes it necessary to deal with the patent clauses even in an only pragmatic OSLIC.

Patent clauses in open source licenses can have two different directions of impact. They use two methods to protect the users of the open source software – and sometimes these methods are combined:

- First, an open source license can assure that all contributors / distributors to / of a piece of open source software grant to all users / recipients not only the right to use the open source software itself, but automatically and implicitly also the right to use all those patents – belonging to the contributors / distributors – which as patents are necessary to use the software legally[209]. So, let us – a little simplifying and therefore only on the following few pages – name such licenses the *granting licenses*.

[206] cf. *Jaeger a. Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software, 2011, p. 235.

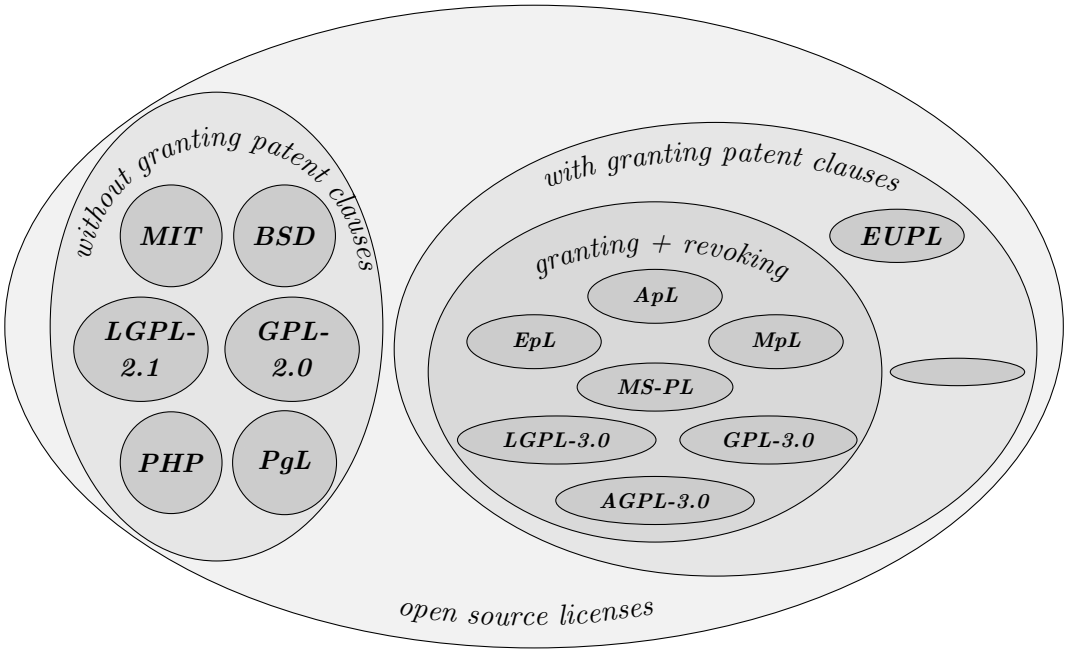
[207] pars pro toto cf. *Open Source Initiative*: APL-2.0, 2004, wp §3.

[208] pars pro toto cf. *Open Source Initiative*: EPL-1.0, 2005, wp.

[209] There might arise a legal discussion whether a distributor who does not contribute to the software development really imlictly also has to grant the necessary rights of his patent portfolio. The OSLiC doesn't want to participate in this discussion. We take a simple and pragmatic position: for being sure that you are acting according to an open source license with such a patent clause you should simply assume that you have to do so. If this default position is not reasonable for you it might be a good idea to consult legal experts which – perhaps – may find another way for you to use the software legally.

- Second, an open source license can try to automatically terminate the right to use, to modify, and to distribute the software if its user initiates litigations against any of the contributors / distributors with respect to an infringement of patent. That can be seen as a revocation of earlier granted rights. So, let us name these license the *revoking licenses*.

Later on, we will summarize the concrete patent clauses of all the licenses discussed in the OSLiC as a proof for the following classification:




But regardless of the final textual form a license is using to express its granting or revoking positions, in any case one has to consider some aspects:

- Overall, one has to keep in mind that of course no licensor, contributor and/or distributor can release the right to use any patents he does not own – even not if he *tries* to release them by an open source patent clause[210]. Implictly touched patents of third parties not having contributed to the

[210] The EPL is one of the licenses which insists on this aspect: It the second half of its patent clause, the EPL underlines that "[...] no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity". Moreover, it explicitly adds that "[...] if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program" (cf. *Open Source Initiative*: EPL-1.0, 2005, wp §2c).

development and/or participated in the distribution can never be implicitly and automatically released on the base of such an (open source) patent clause: no rights, no right to release[211]. Hence: even for those open source licenses which try to protect the users, finally the user itself must nevertheless ensure that he does not violate the patents of third parties being unwillingly touched by the way the code works or the processes in which the software is used[212].

- In the context of a granting license, one has also to consider that contributing to and distributing of a piece of software implicitly evokes that all patents of the contributor and/or distributor are 'given free' which are necessary to use the software as whole – including the more or less deeply embedded libraries. So, if one wants to check whether some of the core patents of one's patent portfolio are afflicted by a patent clause (and whether one therefore better should not use / distribute the corresponding piece of open source software), one should not forget to check the embedded libraries, too.
- Finally, one has to consider in the context of a granting license that its patent clause only releases the use of the patents in the meaning of 'allowed to be used for enabling the use of the distributed software'. The patent clause does not release the patents generally. Thus, the threat of (unwillingly) releasing patents by open source software is not as large as sometimes feared: the use of the patent is only granted in combination with the software. On the one hand, you may not use the open source software without having the right to use the patent because the use of the patent is inherently necessary for using the software – regardless, whether the open source software is embedded into a larger process or not. On the other hand, you are not allowed to use patents – released by the patent clause of an open source license – without exactly that open source software which has been licensed under this open source license, because the patent clause only refers to the use of just that open source software.
- Summarized, one has to consider that the granting open source licenses automatically and implicitly enforce you to grant all the rights which are necessary to use the software legally. Open source contributors and distrib-

[211] This is an important aspect which is sometimes not considered by programmers. Inside of DTAG we had a fruitful discussion evoked by Mr. Stephan Altmeyer who – as patent lawyer – patiently explained this constraint to us.

[212] Sometimes, this problem of willingly or unwillingly violated third party patents is seen as a weakness of open source software. But that is not true. It is a weakness of every software. Even a commercial licensor (developer) has only the right to license the use of those patents he really owns or he has 'bought' for relicensing them. Moreover, also commercial licensors can willingly or unwillingly violate patents of other persons.

utors should know that[213].

- With respect to the revoking licenses, one has to consider that their patent clauses contain negative conditions which may be read as interdictions. The OSLiC will integrate these conditions into specific 'prohibits'-sections of its to-do lists.
- Finally one should mention that in some cases, the form of the revocation used by the revoking license refers to the use of the software, in other cases to the use of the patents. But nevertheless, one can reason that – from the pragmatic viewpoint of a benevolent open source software user – also this second case of patent revocation implicitly terminates the right to use the software: If the use of a patent is necessary to use a piece of software legally, one is not allowed to use the software without having the right to use the patent, too; and if the use of the patent is not necessary for using the software, then the patent is not covered by the patent clause. So, in any case, this kind of patent clauses seems to terminate the right to use / to distribute and/or to modify the software. Hence, single users as well as companies or organizations should also respect such patent clauses if they want to be sure to use open source software compliantly.

The OSLiC wants to support its readers not only to act according to the licenses in general, but also according to its patent clause. Thus, we now briefly cite and summarize the meaning of particular patent clauses:

### 3.1.1 AGPL statements concerning patents [tbd]

### 3.1.2 ApL statements concerning patents

Titled by the headline "Grant of Patent License", the Apache License 2.0 contains a specific patent clause being comprised of two very long and condensed sentences[214]. Outside of this patent clause, the word *patent* is only used once again – for requiring that one "[...] must retain, in the (sources) [...] all [...] patent [...] notices [...]"[215].

The one core message of the ApL patent clause is the statement that "[...] each Contributor hereby grants to You a perpetual, worldwide, non-exclusive,

213) Again: It might be debatable whether this is also valid for the distributors which do not contribute anything to the development. That's a legal discussion the OSLiC do not wish to participate in. From the viewpoint of an open source user who only wants to have one reliable and secure way to use open source software compliantly, one should perhaps assume that there is no difference.

214) cf. *Open Source Initiative*: APL-2.0, 2004, wp §3.

215) cf. id., l.c., wp §4.3.

no-charge, royalty-free, irrevocable [...] patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work [...]"[216].

The second core message of the ApL patent clause is the statement that "if You institute patent litigation against any entity [...] alleging that the Work [...] constitutes [...] patent infringement, then any patent licenses granted to You [...] shall terminate [...]"[217].

The third message of the ApL patent clause is the statement, that the "[...] license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted"[218].

Thus, the ApL is – as we are using to say in this chapter – a granting and a revoking license: At first you are granted to use all patents of all contributors which are necessary to use the software legally. But if you – with respect to the software – install any litigation concerning an infringement of patents, then the rights granted to you are revoked.

### 3.1.3 CDDL statements concerning patents [tbd]

[...]

### 3.1.4 EPL statements concerning patents

The Eclipse Public License treats the patents being necessary to use the program in the same section and under the same headline "Grant of Rights" like all the other rights: First, the EPL clearly states that "[...] each Contributor [...] grants (the recipient) a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form"[219]. Then the EPL delimits the extend of this act of granting: Neither hardware patents of the contributors are covered by this releasing patent clause, nor patents that concern aspects out of the area of the initially intended software combination[220]. Finally, the EPL hints to the general fact that 3rd party patents not belonging to the contributors can never be implicity be released by such a

[216] cf. *Open Source Initiative*: APL-2.0, 2004, wp §3. The "Contributor", "Work" and "You" are defined in §1: *Contributor* refers to the original licensor and to all others whose contributions have been incorporated into the Work. The *Work* denotes the result of the development process regardless of its form. *You* denote the licensees.

[217] cf. id., ibid.

[218] cf. id., ibid.

[219] cf. *Open Source Initiative*: EPL-1.0, 2005, wp §2.b.

[220] cf. id., ibid.

patent clause. Moreover, it alleges the example that "[...] if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program"[221].

Like other open source licenses, also the EPL announces at its end that "if (a) Recipient institutes patent litigation against any entity [...] alleging that the Program [...] infringes such Recipient's patent(s), then such (granted) Recipient's rights [...] shall terminate [...]"[222].

Thus, also the EPL is – as we are using to say in this chapter – a granting and a revoking license: At first you are granted to use all patents of all contributors which are necessary to use the software legally. But if you – with respect to the software – install any litigation concerning an infringement of patents, then the rights granted to you are revoked.

### 3.1.5 EUPL statements concerning patents

The European Union Public License contains a very brief patent clause. It 'only' states, that "the Licensor grants to the Licensee royalty-free, non exclusive usage rights to any patents held by the Licensor, to the extent necessary to make use of the rights granted on the Work under this Licence"[223]. Furthermore the EUPL does not contain any patent specific revoking clause, but only an abstract clause requiring that all "[...] the rights granted hereunder will terminate automatically upon any breach by the Licensee of the terms of the Licence"[224]. Thus, the EUPL is – as we are using to say in this chapter – a granting license and not a revoking license.

### 3.1.6 GPL statements concerning patents

Although the GPL versions 2.0 and 3.0 are aiming for the same results, they heavily differ with respect to textual and arguing structure. Therefore, it should be helpful to treat these two licenses separately.

#### 3.1.6.1 GPL-2.0

The GPL-2.0 does not contain any specific patent clause by which it would grant (and revoke) the rights to use those patents belonging to the contributors and being necessary to use the software in accordance with the legal patent system.

221) cf. *Open Source Initiative*: EPL-1.0, 2005, wp §2.c.
222) cf. id., l.c., wp §7.
223) cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp. §2 at its tail.
224) cf. id., l.c., wp. §12.

Instead of this, the preamble of the GPL-2.0 alleges that "[...] any free program is threatened constantly by software patents" and that the authors of the GPL – for tackling this threat – "[...] had made it clear that any patent must be licensed for everyone's free use or not licensed at all"[225]. Unfortunately, this specification is only an indirect claim which needs a lot of arguing for establishing a protective effect against patent disputes. Howsoever, this paragraph of the GPL-2.0 does not directly grant any rights to the software users to use necessary patents, too.

With respect to the patent problem, the GPL-2.0 also states that a licensee has to fulfill the conditions of the GPL-2.0 completely, even if an existing patent infringement – being "imposed" on the GPL licensee – "[...] contradicts the conditions of this license" so, that a waiver of the use of the software is the only way to fulfill both constraints[226]. And finally the GPL-2.0 allows the original copyright holder to "add an explicit geographical distribution limitation excluding [...] countries" provided that these countries "[...] (restict) the distribution and/or use of the library [...] by patents [...]"[227]. Based on these statements, one cannot infer that the GPL-2.0 grants any patent rights to the software user, neither directly, nor indirectly.

Thus, the GPL-2.0 is not a granting or a granting and revoking license.

#### 3.1.6.2 GPL-3.0

Initially, the GPL-3.0 regrets that "[...] every program is threatened constantly by software patents" what should be seen as the "[...] danger that patents applied to a free program could make it effectively proprietary". And therefore – as the GPL-3.0 itself summarizes its patent rules – "[...] the GPL assures that patents cannot be used to render the program non-free"[228]. This kind of protection is then established by three steps. First, the GPL-3.0 stipulates that "each contributor grants [...the licensees] a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version"[229]. Second, the GPL-3.0 defines that this patent license granted by the contributor "[...] is automatically extended to all recipients" who later on receive any version of the work, even if they indirectly receive them by third parties and even if they receive a "covered work" or "works based on it"[230]. Moreover, the GPL-3.0 also specifies that those distributors of a "covered work"

225) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp, Preamble.
226) cf. id., l.c., wp. §11.
227) cf. id., l.c., wp. §12.
228) cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. Preamble.
229) cf. id., l.c., wp. §11.
230) cf. id., ibid.

who have the right to use a patent necessary for the use of the distributed software but who are not allowed to relicense this patent to third parties must solve this problem by making the source code available nevertheless, by "depriving" themselves or by "extending the patent license to downstream recipients"[231]. And finally, the GPL-3.0 also introduces a revoking clause by stating that a licensee "[...] may not initiate litigation [...] alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it"[232] and that this licensee "automatically" loses the rights granted by the GPL-3.0 – "including any patent licenses" – if he tries to propagate or modify a covered work against the rules of the GPL-3.0[233].

Thus, GPL-3.0 is – as we are using to say in this chapter – a granting and a revoking license: At first, one is granted the right to use all patents of all contributors which are necessary to use the software legally. But if you – with respect to the software – install any litigation concerning an infringement of patents, then the rights granted to you are revoked.

### 3.1.7 LGPL statements concerning patents

As already mentioned above, the LGPL versions 2.1 and 3.0 heavily differ with respect to textual and arguing structure. Therefore, they should be treated separately.

#### 3.1.7.1 LGPL-2.1

Like the GPL-2.0, the LGPL-2.1 does not contain any specific patent clause by which it would grant (and revoke) the rights to use those patents belonging to the contributors and being necessary to use the software in accordance with the legal patent system.

Instead of this, the preamble of the LGPL-2.1 says that "[...] software patents pose a constant threat to the existence of any free program" and that the authors of the LGPL – for tackling this threat – "[...] insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license"[234]. Unfortunately, this specification is again only an indirect claim which needs a lot of arguing for establishing a protective effect against patent disputes. Howsoever, this paragraph of the LGPL-2.1 does not directly grant any rights to the software users to use necessary patents too.

[231] cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §11.
[232] cf. id., l.c., wp. §10.
[233] cf. id., l.c., wp. §8.
[234] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp, Preamble.

With respect to the patent problem, the LGPL-2.1 also states that a licensee has to fulfill the conditions of the LGPL-2.1 completely, even if an existing patent infringement – being "imposed" on the LGPL licensee – "[...] contradicts the conditions of this license" so that a waiving of the use of the software is the only way to fulfill both constraints[235]. And finally the LGPL-2.1 allows the original copyright holder to "add an explicit geographical distribution limitation excluding [...] countries" provided that these countries "[...] (restict) the distribution and/or use of the library [...] by patents [...]"[236]. Based on these statements, one cannot infer that the LGPL grants any patent rights to the software user, neither directly, nor indirectly.

Thus, also the LGPL-2.1 is not a granting or a granting and revoking license.

#### 3.1.7.2 LGPL-3.0

The LGPL-3.0 is an extension of the GPL-3.0. Before starting with a section "Additional Definitions", the LGPL-3.0 states that it "[...] incorporates the terms and conditions of version 3 of the GNU General Public License" and then "supplements" this GPL-3.0 content by some "additional permissions"[237]. The LGPL-3.0 itself does not contain the word 'patent', but the GPL-3.0[238]. So, the LGPL-3.0 inherits its patent clause from the GPL-3.0 which is – as we already described[239] – a granting and a revoking license.

### 3.1.8 MPL statements concerning patents

The MPL distributes its statements concerning the tolerated use of the patents over three paragraphs: First, it clearly says that "each Contributor [...] grants [...the licensee] a world-wide, royalty-free, non-exclusive license [...] under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version"[240]. Second, it hihlights some "limitations"[241]. And finally, the MPL introduces a revoking clause which signifies that the rights, granted to the licensee "[...] by any and all Contributors [...] shall terminate" if the licensee "initiates litigation against any entity by asserting a patent infringement claim [...]

[235] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §11.
[236] cf. id., l.c., wp. §12.
[237] cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp.
[238] cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §11.
[239] → OSLiC, p. 54
[240] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §2.1, esp. §2.1.b.
[241] cf. id., l.c., wp. §2.1, esp. §2.3.

alleging that a Contributor Version directly or indirectly infringes any patent [...]"[242].

Thus, the MPL is – as we are using to say in this chapter – a granting license and a revoking license.

### 3.1.9 MS-PL statements concerning patents

First, the MS-PL contains a statement, by which "[...] each contributor grants (the software users) a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software"[243]. Second, the MS-PL says that "if you bring a patent claim against any contributor[...] your patent license from such contributor to the software ends automatically"[244].

Thus, the MS-PL is – as we are using to say in this chapter – a granting and a revoking license: At first you are granted to use all patents of all contributors which are necessary to use the software legally. But if you – with respect to the software – install any litigation concerning an infringement of patents, then the rights granted to you are revoked.

## 3.2 Excursion: Why linking is a secondary criteria

> *Distributing statically or dynamically linked software is often discussed as a problem (and sometimes as a solution) for acting compliantly. In this chapter, we briefly discuss why this aspect can mostly be ignored and why it does not help to determine the existence of a derivative work.*

In some earlier versions of the OSLiC, its finder subclassified some use cases with respect to the way an application was 'composed' as a larger unit: In the previous form for gathering the necessary information, the OSLiC user had to answer whether he *was going to combine the received open source software with other software components by linking all together statically, by linking them dynamically, or by textually including (parts of) the open source software into [his] larger unit.* Today, this question has totally been erased. The authors could convince themselves that it is not necessary to consider this aspect.

Of course, we know that being linked statically or dynamically is often and deeply

242) cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §5.2.
243) cf. *Open Source Initiative*: MS-PL, 2013, wp §2.B.
244) cf. id., l.c., wp §3.B.

discussed by license experts[245]. It seems to be an important aspect:

[TBD: Discussion of the literature]

So, let us start with some undeniable facts: The OSLiC deals with the Apache-2.0 license[246], the BSD-2-Clause license[247], the BSD-3-Clause license[248], the MIT license[249], the MS-PL[250], the PgL[251] and the PHP license[252] as instances of the permissive licenses. Additionally, the OSLiC treats the EPL[253], the EUPL[254], the LGPL[255], and the MPL[256] as licenses with weak copyleft. Finally, the OSLiC thoroughly discusses the GPL[257] and the AGPL[258] as licenses with strong copyleft[259].

Only three of these licenses mention the word *linking* (or variants of it): Using the command `grep -i link * | grep -v "<link\|links\|skip-link"` in a shell – executed as an operation on a set of html formatted license files – directly shows that only the AGPL-3.0, the ApL-2.0, the GPL-2.0, the GPL-3.0, the LGPL-2.1 and the LGPL-3.0 are using mutations of the word *linking*. Additionally, the results of the command `grep -i statical *` show that only the LGPL-2.1 uses the word 'statical', while using the command `grep -i dynamical *` only hints to the AGPL-3.0 and the GPL-3.0. Finally, the command `grep -i "shared" *` – executed on the same set of files – shows that the term *shared libary* is also only used by these licenses.

This analysis already indicates that being statically or dynamically linked might not be as important for acting compliantly as it is often suggested. If one reads the concrete statements, then one can see, that acting compliantly depends only

245) Even on the *European Legal and Licensing Workshop, 2013* in Amsterdam, there was given an excellent lecture concerning the nature and concequences of linking elf files.
246) cf. *Open Source Initiative*: APL-2.0, 2004, wp.
247) cf. *Open Source Initiative*: The BSD 2-Clause License, 2012, wp.
248) cf. *Open Source Initiative*: The BSD 3-Clause License, 2012, wp.
249) cf. *Open Source Initiative*: The MIT License, 2012, wp.
250) cf. *Open Source Initiative*: MS-PL, 2013, wp.
251) cf. *Open Source Initiative*: PostgreSQL License, 2013, wp.
252) cf. *Open Source Initiative*: PHP-3.0, 2013, wp.
253) cf. *Open Source Initiative*: EPL-1.0, 2005, wp.
254) cf. *Open Source Initiative*: EUPL-1.1 (OSI), 2007, wp.
255) For LGPL-2.1 see cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. For LGPL-3.0 see cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp
256) cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp.
257) For GPL-2.0 see cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. For GPL-3.0 see cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp
258) cf. *Open Source Initiative*: GNU Affero General Public License, Version 3 (AGPL-3.0); 2007 [n.y. of the html page itself] ⟨URL: `http://opensource.org/licenses/AGPL-3.0`⟩ – reference download: 2013-04-05, wp.
259) You can find html based instances of these licenses in the OSLiC directory 'licenses'. They have been downloaded from the OSI pages. All of the following statements refer to these files.

slightly and only rarely on the kind of being 'combined':

**ApL-2.0:** This version of the Apache license uses the word *link* only once for stating that "[...] Derivative Works shall not include works that [...] link [...] to the interfaces of, the Work and Derivative Works thereof"[260]. Thus, the ApL does not use the criteria *being linked* for determining a derivative work, neither *being linked* in general, nor *being statically linked*, nor being *dynamically linked*. Hence, for acting in accordance to the ApL, this class of attributes can completely be ignored.

**GPL-3.0:** The GPL-3.0 uses the word *link* three times: First, it defines the "'Corresponding Source' for a work in object code form [...as] all the source code needed to generate, install, and [...] run the object code and to modify the work [...]". Additionally the GPL-3.0 also explains in this context that this definition shall include "[...] the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require"[261]. Second, the GPL-3.0 allows "[...] to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work"[262]. Finally, the GPL-3.0 explains that "the GNU General Public License [itself] does not permit incorporating your program into proprietary programs" and that the LGPL might be a better license for those licensors who have written a "subroutine library [...] and may consider it more useful to permit linking proprietary applications with the library [...]"[263].

So, also in this text, the features *statically linked* or *dynamically linked* are not used to trigger any license fulfilling actions. The conditions for "Conveying Modified [...] Versions" refer to the "work based on the Program"[264] which itself denotes a "'modified version' of the earlier work"[265]. Moreover, the licensee – as modifier, distributor, and subsequent licensor – is required by the GPL-3.0 "[...] to license the entire work [which has been developed on the base of a GPL-3.0 component], as a whole, under this License to anyone who comes into possession of a copy"[266]. The GPL-3.0 does not limit this claim – especially not by referring to a mode of being linked. Hence, also with respect to the GPL-3.0, one can completely ignore these features of the software, its use and its distribution for determining how to use the software compliantly.

260) cf. *Open Source Initiative*: APL-2.0, 2004, wp. §0.
261) cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §0.
262) cf. id., l.c., wp. §13.
263) cf. id., l.c., wp. last parapgraph.
264) cf. id., l.c., wp. §5.
265) cf. id., l.c., wp. §0.
266) cf. id., l.c., wp. §5.

**AGPL-3.0:** Concerning the use and the meaning of the words *dynamically* and *linking*, the AGPL-3.0 exactly follows the structure of the GPL-3.0: first the terms arise in the context of defining the "Corresponding Source"[267]; then the word *link* helps to say that AGPL and GPL are compatible licenses[268]; and finally the word *link* is used to hint to the LGPL[269]. So, again, one can ignore the feature of being statically or dynamically linked if one wants to determine how to use the software compliantly.

**GPL-2.0:** In the GPL-2.0, the word *link* only arise in the context of hinting to the LGPL[270]. Moreover, the words *statical* and *dynamical* are not used in this text – not at all and in no sense: the copy left feature of the GPL depends 'only' on a specification which refers to a "work based on the Program [...] that in whole or in part contains or is derived from the Program or any part thereof [...]"[271]. Thus, even in this old version of the GPL, the criteria of being linked – in which way ever – does not trigger any task for using the software compliantly.

**LGPL-3.0:** In this license, variants of the word *link* are used to define the concept of a "Combined Work" which shall be the name for a "[...] work produced by combining or linking an Application with the Library"[272]. In the end the LGPL-3.0 allows to "[...] convey a Combined Work under terms of [his own] choice [...]", provided that one distributes also all material (including the object files of the overarching on-top developments) being necessary for enabling the receiver to relink the whole product with a later incoming newer version of the library or that one presupposes the use of a "suitable shared library mechanism" so that the receiver can update the library simply by replacing the binary library file[273]. For fulfilling these conditions it is sufficient to require that a distributor shall *either distribute the on-top development and the library in the form of dynamically linkable parts or distribute the statically linked application together with a written offer, valid for at least three years, to give the user all object-files of the on-top development and the library, so that he can relink the application on its own behalf.*

**LGPL-2.1:** Even if the LGPL-2.1 is more sophistically arguing than all the other licenses, in its preamble this license clearly states what it wants to evoke: "If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after

267) cf. *Open Source Initiative*: The AGPL-3.0 License (OSI), 2007, wp. §0.
268) cf. id., l.c., wp. §13.
269) cf. id., l.c., wp. §5.
270) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. last paragraph.
271) cf. id., l.c., wp. §2.
272) cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp. §0.
273) cf. id., l.c., wp. §4.

making changes to the library and recompiling it [...]"[274]. For that purpose, the LGPL-2.1 defines in the beginning that if "a program is linked with a library, whether statically or using a shared library, [then] the combination of the two is legally speaking a combined work, a derivative of the original library"[275]: On the one hand a "work that uses the Libary" – which is only "[...] designed to work with the Library by being compiled or linked with it [...]" – "[...] in isolation, is not a derivative work of the library [...]". On the other hand, it is no question for the LGPL-2.1, that "linking a 'work that uses the Library' with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library)"[276]. But then – "as an exeption" – the LGPL-2.1 allows to "[...] combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice". The right to do this is granted provided that the distributor either presupposes the use of a "suitable shared library mechanism" or that he distributes also the complete material (including the object files of the overarching on-top developments) which is necessary to enable the receiver to relink the whole product with a later incoming newer version of the library[277]. Again, for fulfilling all these conditions it is sufficient to require that a distributor shall *either distribute the on-top development and the library in the form of dynamically linkable parts or distribute the statically linked application together with a written offer, valid for at least three years, to give the user all object-files of the on-top development and the library, so that he can relink the application on its own behalf.*

Thus, with respect to this analysis, we can conclude that – in general – there is no need to gather more or less complicately whether one wants to distributed software in the form of statically or dynamically linked binaries for deriving the necessary tasks to distribute this software compliantly. Instead of this, we can directly incorporate those doings into the task lists of the LGPL what has been discovered as sufficient doings. Moreover, it is also sufficient to insert this statement only in the task list of the LGPL. There is no need to generalize this discussion. So, we could simplify our form offered to gather the information to find the adequate license fulfilling task list.

274) cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. preamble.
275) cf. id., ibid.
276) cf. id., l.c., wp. §5.
277) cf. id., l.c., wp. §6, §6b and §6c together with §6c.

## 3.3 Excursion: What is a 'Derivative Work' - the basic idea of open source

*This chapter briefly discusses existing attempts to define the derivated works of technical aspects, like dynamical or statical linking or not. We will prove that linking can not deliver a definite criteria: 1) modules are only unzipped libraries. 2) you can distribute software as modules added by a script, which statically(sic!) links all modules before executing the program. 3) The criteria of pipe-communication is good, but not sufficient. 4) All these attempts do not match the constituting features of script languages. Therefore we will follow Moglen(?) and will argue from the viewpoint of a developer: it is only a question of a function, method or anything else which calls (jumps into) a piece of code which has been licensed by a license protecting on-top-developments and you have a derivated work.*

...

IMPORTANT*: This text is only a first draft which sketches some aspects. Later on we will fully elaborate the complete argumentation.*

As first version, we present an outline of the arguing structure this chapter will later part of a more narrative way.

**The meaning 'derivative work' must be known!** Many open source licenses are using the the term 'derivative work' [cite the sources], either direct or indirect in form of the work 'modification' [Write a table as survey]. And nearly all licenses, which are using the term 'derivative work' etc., are linking tasks which must be executed to comply with the corresponding license, to the precondition, that something is a derivative work. [table survey]. **Hence, for acting in accordance to such a license, it has to be known, what a derivate work is**

**Unfortunately the meaning is not clearly fixed** . The exist some different readings of the term 'derivative work' [specify the differences and cite the sources] **Hence, it is not as clear as wished what a derivative work is**

**Hence lets argue from the viewpoint of a benevolent developer** : Open source licenses are written for software developers, mostly to preserve their freedom, to develop software. And sometimes these licenses are also written by software developers – or at least by the assistance of. So, one should be able to answer the question under which circumstances a piece of software is a 'derivative work' of another piece of software on the base of two principles:

- Let us argue on the base of a benevolent neutral software developer without hidden interests or a hidden agenda.
- In case of doubts let us preferably assume that the two pieces interrelate as source and derivative work – so that the OSLiC rather

recommends to execute the required tasks than to put them away.

Basically we generalize a specific viewpoint of the LGPL: It uses three terms:

**"library"** is defined as "a collection of software functions and/or data prepared so as to be conveniently linked with application programs"[278].

**"work based on the library"** is defined as "either the library or any derivative work"[279].

**"work that uses the library"** is defined as something which initially "[...] is not a derivative work of the library [...]" but can become a derivative work by being combined / linked to the library it uses[280].

Following these specifications, one has to conclude that there can be derived *derivative works* of the library in two different ways: First, the library itself can be enhanced without changing the character of being a library. Then, of course, the resulting library is a derivative work of the initial library. Second, an overaching program can use the library by calling functions, methods or data, offered by the library. In this case, the overarching program functionally depends on the library and is a derivative work (as soon as it is linked to the library).

This viewpoint can be generalized: also snippets, modules, plugins can be enhanced and used by overarching programs or even by more complex libraries. Based on this viewpoint - which should finally be formulated as the viewpoint of a benevolent impartial developer - the OSLiC uses the following rules by which the OSLiC decides to take something as derivative Work:

**Copy-Case** Copying a piece of code from a source file and pasting it into a target file makes the target file a derivatve work of the source file[281].

**Modify-Case** Inserting any new content or deleting any existing content of a source file makes the resulting target file being a derivate work of the source file.

**Call-Case** Inserting into a target file the call of function which is defined inside of and delivered by a sourcefile makes the target file depending on the source file and therefore a derivative work of the delivering source file.

And here are some applications of the rules

278) cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp §0.
279) cf. id., ibid.
280) cf. id., l.c., wp §5.
281) Be careful: this case must still be distinguished from the case of an automatically inclusion (header files, script libraries) during the compilation / execution: Header files allon should not evekoe a derivative work.

- Combining the copy-case and the modify-case on the base of two different source files make the resulting target file being a derivative work of the two source files (follows from copy-case and modify-case)
- ...

[... TBD ...]

## 3.4 Excursion: The problem of license compatibility [tbd]

*Here we discuss the often neglected or only loosely touched problem of combining differently licensed software. We will hint to the Exclusion-List of the Free software foundation; we will hint to the Eclipse / GPL-plugin problem; we will mention the recent discussion whether the kernel requires to license the complete Android as GPL; and finally we will discuss the just now published, short analysis of Jaeger and Metzger presenting a combining matrix which seems to fall into their lap. We ourselves will argue that the question can simply be answered: only if you embed two libraries which both are licensed by an on-top-development protecting license and if these both licenses require the licensing of the derivated work by different licenses then you have a problem. In all other cases which we will describe there is no problem.*

...

## 3.5 Excursion: open source software and money [tbd]

*Here we will shortly discuss ways in which money and Open Source is no problem.*

...

# 4 Open Source Use Cases: Concept and Taxonomy

> *This chapter establishes our concept of* open source use cases *as a classification system for to-do lists. The conditions of a specific license, in the context of a particular* open source use case*, shall be satisfiable by following the corresponding to-do list. Additionally this chapter introduces a taxonomy for these* open source use cases*. Later on, this taxonomy will organize the* Open Source Use Case Finder.

After all these introductory remarks, we can summarize our idea. We know that the right to use open source software depends on the tasks required by the open source licenses. As opposed to commercial licenses, you can not buy the right to use a piece of open source software by paying money. It is embedded into the *Open Source Definition* that the right to use the software may not be sold. The OSD states firstly that an open source license may "[...] not restrict any party from selling or giving away the software as a component of (any) aggregate software distribution", and adds secondly in the same context that an open source license "[...] shall not require a royalty or other fee for such sale"[282].

However, it would be wrong to conclude that you are automatically allowed to use open source software without any service in return: generally you have to do something to gain the right to use the software. In other words: open source software is covered by the idea of 'paying by doing'. Accordingly, open source licenses describe specific circumstances under which the user must execute some tasks in order to be compliant with the licenses. So, if we want to offer to-do lists for fulfilling license conditions, we must consider these tasks and circumstances.

In practice, such circumstances are not linear and simple. They contain combinations of (sometimes context sensitive) conditions which can be grouped into classes of tokens. Such a class of tokens might denote a feature of the software itself – such as being an application or a library. Or it can refer to the circumstances of using the software, such as 'using the software only for yourself' or 'distributing the software also to third parties'.

At the end, we want to determine a set of specific OSUCs – the *open source use cases*. And we want to deliver for each of these OSUCs and for each of the considered open source licenses one list of actions which fulfills the license in that context[283].

Such an *open source use case* shall be considered as a set of tokens describing the

[282] cf. *Open Source Initiative*: The Open Source Definition, 2012, wp §1.

[283] Fortunately, sometimes one task list fulfills the conditions of more than one use case – a welcome reduction of complexity

circumstances of a specific usage. Hence, to begin, we must specify the relevant classes of tokens, before we can determine the valid combinations of these tokens – our *open source use cases*. Finally, based on the tokens, we generate a taxonomy in the form of a tree. This tree will become the base of the *Open Source Use Case Finder* which will be offered in the next chapter, and which leads you to your specific OSUC by evaluating just a few questions and answers.

There are only a handful of tokens which are relevant to the circumstances of open source software licenses:

- The **type of the open source software**: On the one hand, we regard code snippets, modules, libraries and plugins, and on the other hand, autonomous applications, programs and servers. We will take the word 'snimolis' for the first set, and 'proapses' for the second. This is necessary, as we are not only talking about libraries and applications in the everyday sense, but rather in the broadest sense[284]. More specifically, we will ask you, whether the open source software you want to use, is an includable code snippet, a linkable module or library, or a loadable plugin, or whether it is an autonomous application or server which can be executed or processed. In the first case, the answer should be 'it is a snimoli', in the second 'it is a proapse'.
- The **state of the open source software**: It might be used exactly as one has received it. Or it can be modified, before being used. More specifically, we will ask you, whether you want to leave the open source software as you have received it, or whether you want to modify it before using and/or distributing it to 3rd parties. In the first case, the answer should be 'unmodified', in the second 'modified'.
- The **usage context of the open source software**: On the one hand you might use the received open source software as a readily prepared application. On the other hand you might embed the received open source into a larger application as one of its components. More specifically, we will ask you, whether you are using the open source software as an autonomous piece of software, or whether you are using it as an embedded part of a larger, more complex piece of software. In the first case, the answer should be 'independent', in the second 'embedded'.
- The **recipient of the open source software**: Sometimes you might wish to use the received open source software only for yourself. In other cases

[284] Of course, our newly introduced concepts of 'snimoli' and 'proapse' are not absolutely one of the most elegant words. So, initially we tried to talk about 'applications' and 'libraries', although in our context these words should denote more, than they traditionally do. But we couldn't minimize the irritations of our interlocutors. Too often we had to remind them that we were not talking about applications and libraries in the strict sense of the words. Finally we decided to find our own words – and to stay open for better proposals ;-)

you might intend to hand over the software (also) to other people. More specifically, we will ask you, whether you are going to use the open source software only for yourself, or whether you plan to (re)distribute it (also) to third parties. In the first case, the answer should be '4yourself', in the second '2others'.

- The **form of the distributed files**: Many licenses also draw a distinction between distributing the software as sources and distributing the files as binaries. In this case, we will ask you, whether you want to distribute the software in the form of binaries or as source code. In the first case, the answer should be 'binaries', in the second 'sources'

From a more programmatic point-of-view, we can summarize these tokens as follows:

- `type::snimoli` *or* `type::proapse`
- `state::unmodified` *or* `state::modified`
- `context::independent` *or* `context::embedded`
- `recipient::4yourself` *or* `recipient::2others`
- `form::binaries` *or* `form::sources`

We have already defined the *open source use case* as the combination of these tokens. If we simply combine all these tokens of all these classes with all the tokens of the other classes[285], we get 2*2*2*2*2 = 32 sets of tokens – or 32 *open source use cases*. Fortunately, some of the generated sets are invalid from an empirical or logical view, and some of these sets are context sensitive:

1. It would be unreasonable to ask you whether you are going to combine the received software with other software components by linking them statically or dynamically, or by including it textually into a larger unit, if you already have answered that the received open source software is a *proapse* or that it shall be used *independently*: A readily prepared application or server can't be linked to another application or server which also contains a `main`-function. And using a *proapse* or *snimoli independently* implies that it is not used *in combination* with other units.
2. If you already have specified that the used open source software is a *proapse* – an autonomous program, an application, or a server – then your answer

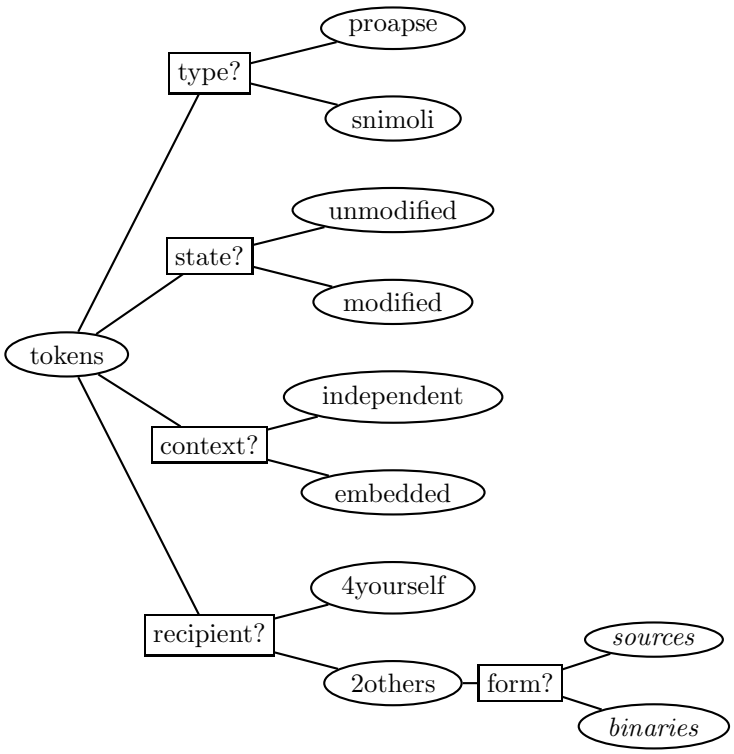
[285] in the sense of the cross product TYPE × STATE × CONTEXT × RECIPIENT × FORM. In some earlier versions of the OSLiC, we also asked whether you are going to combine or to embed the open source software with other software components by linking them statically or dynamically, or by textually including (parts of) the open source software into your larger product. Meanwhile, we clearly discovered that it is unnecessary to increase the complexity by the results of this question. For Details → OSLiC p. 57

implies that the software is used independently and is not embedded with other components into a larger unit. But if you have specified that the used open source software is a *snimoli* – a snippet of code, a module, a plugin, or a library – then it can indeed be used as an embedded component of a constructed larger application or server, or it can be used independently in case you 'only' re-distribute it to 3rd. parties.

3. If you already have specified that the used open source software is a *snimoli* – a snippet of code, a module, a plugin, or a library – and that this *snimoli* shall be used only by yourself (not distributed to other 3rd. parties) then your answer must also imply that this *snimoli* is used in combination, as an embedded part of a larger unit. A library can not be used autonomously, without using it as a component of another application. In this case, it would simply sit on the disk and would do nothing more than occupying space.

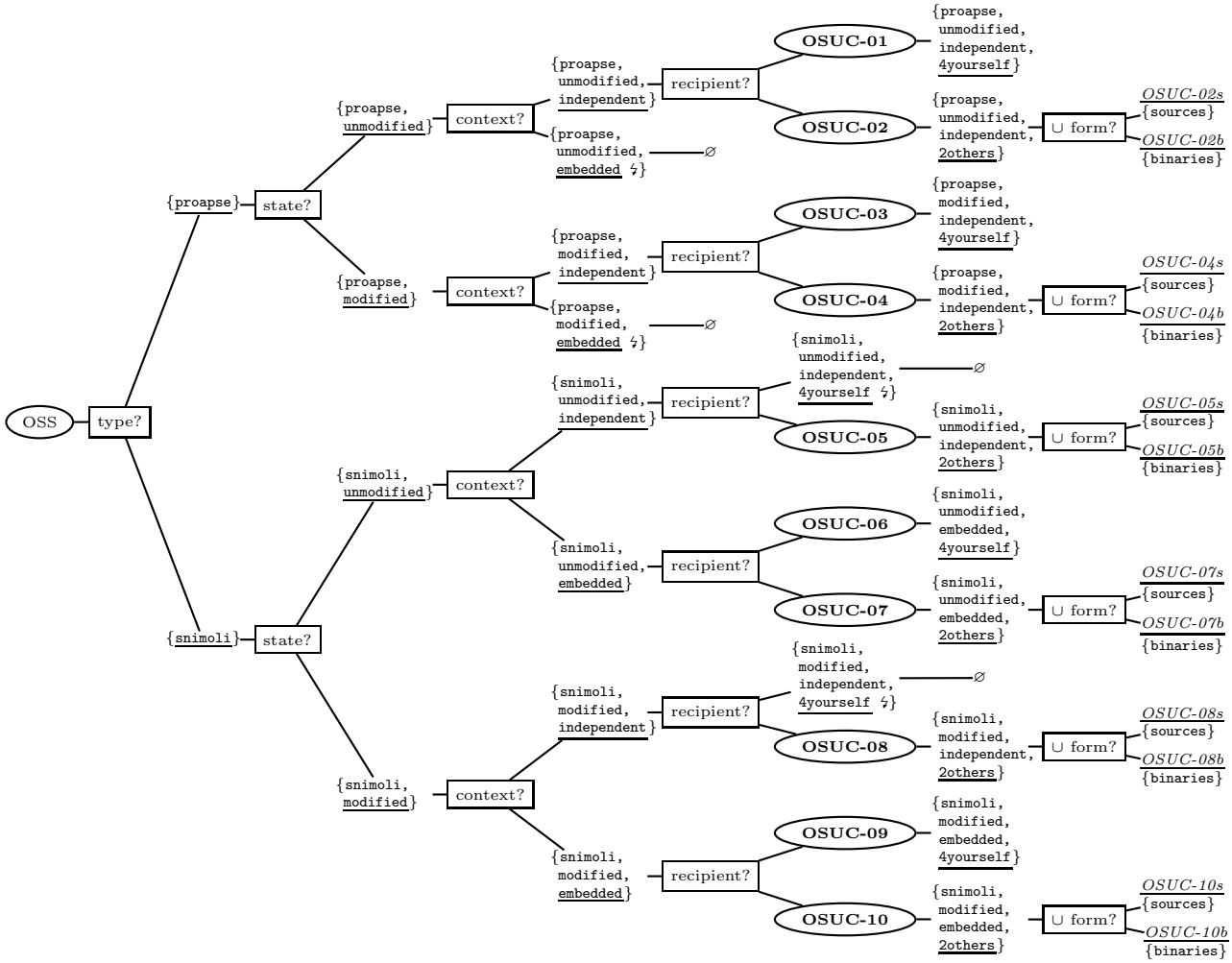
Does this sound complex? We thought so, too. We spent much time explaining these constraints to ourselves, and only when we had transposed all the combinations and rules into a tree, the situation became clearer. The following diagrams shall summarize our own clarifications:

## 4.1 Overview of the OSUC classes and tokens



## 4.2 The OSUC taxonomy

This is one of the possible trees 'collecting' the tokens and offering the *open source use cases* as their leafs[286]:




[286] Each of the invalid use cases (= sets of tokens) [for details s. p. 67] is marked by an ↯ and leads to an empty set (= ∅): A proapse can not be embedded with another software unit, also containing a main-function. Using a software library only for yourself and independent (not in combination with larger software unit), is like having an unused heap of bytes on your disc.

# 5 Open Source Use Cases: Find the License Fulfilling To-do Lists

*This chapter offers the* Open Source Use Case Finder*: Based on the information gathered by a form, it allows to traverse a tree whose leafs are linked to the* open source use cases *which finally refer to the respective to-do lists.*

## 5.1 A standard form for gathering the relevant information

| | *Which open source software do you want to use?* | |
|---|---|---|
| | *Under which open source license is it released?* | |
| **Focus** | **Questions** | **Answers** |
| Type | *Is the open source software you want to use a library in the broadest sense (an includable code **sni**ppet, a linkable **mo**dule or **li**brary, or a loadable plugin), or is it an autonomous **pro**gram, **ap**plication or **se**rver which can be executed?* | ☐ snimoli<br>☐ proapse |
| State | *Do you want to leave the open source software **unmodified** as you have received it, or are you going to create a **modified** version of it?* | ☐ unmodified<br>☐ modified |
| Context | *Are you going to use / distribute the open source software as an **independent** unit, or do you plan to integrate it as an **embedded** component into a complexer piece of software?* | ☐ independent<br>☐ embedded |
| Recipient | *Are you going to use the open source software only **for yourself**, or do you plan to (re)distribute it (also) **to other** third parties?* | ☐ 4yourself<br>☐ 2others |
| Form | *Given you want to (re)distribute an open source based work [2others], do you focus on distributing the **binaries** or the **sources**?* | ☐ binaries<br>☐ sources |

As discussed earlier, there are of course some invalid combinations[287]. Here are some extra explanations concerning the classes resp. the focuses:

[287] type::proapse excludes state::embedded; recipient::4yourself excludes the combination with state::independent and type::snimoli; any value of class 'mode' implies state::embedded [for details see page 67]. If you have gathered one of these invalid combinations, please check the corresponding explanations

**Type:** A piece of (open source) software is a program, an application, or a server, only if you can start its binary form with your normal program launcher, or (in case of a text file which still must be interpreted by an interpreter like php, perl, bash etc.) if you can start an interpreter which takes the file as one of its arguments and executes the commands.

**State:** You are modifying a piece of (open source) software if you expand, reduce or modify at least one of the received software files, and – in case of dealing with binary object code – if you (re)compile and (re)link the modified software to a new binary file. But if you only modify some of the configuration files, you are not modifying the open source software itself.

**Context:** You are using a piece of open source software as an embedded component of a larger unit ...

- if one of your files of the larger unit contains a verbatim or a modified copy (i.e. a snippet) of the received open source software, or
- if your larger unit contains an include statement referring to a functionally defining file of the received open source software, or
- if your larger unit calls a function defined in the received open source software, or
- if your development environment contains a compiler or linker directive referring to the received open source software (binaries) and if your larger unit can't be executed without resolving this linker directive.

**Recipient:** You are using the received open source software only for yourself, if you as a person do not pass it to other entities like persons, organizations, companies etc., or if you – as a member of a specific development group – pass it only to the other members of your development group. But if you store open source software on any device such as a mobile phone, an USB stick, etc. or if you attach it to any transport medium like email etc. and if you then sell, give away, or simply send this device or transport medium to anyone (other than a direct member of your development group) then you indeed handover the open source software to third parties[288].

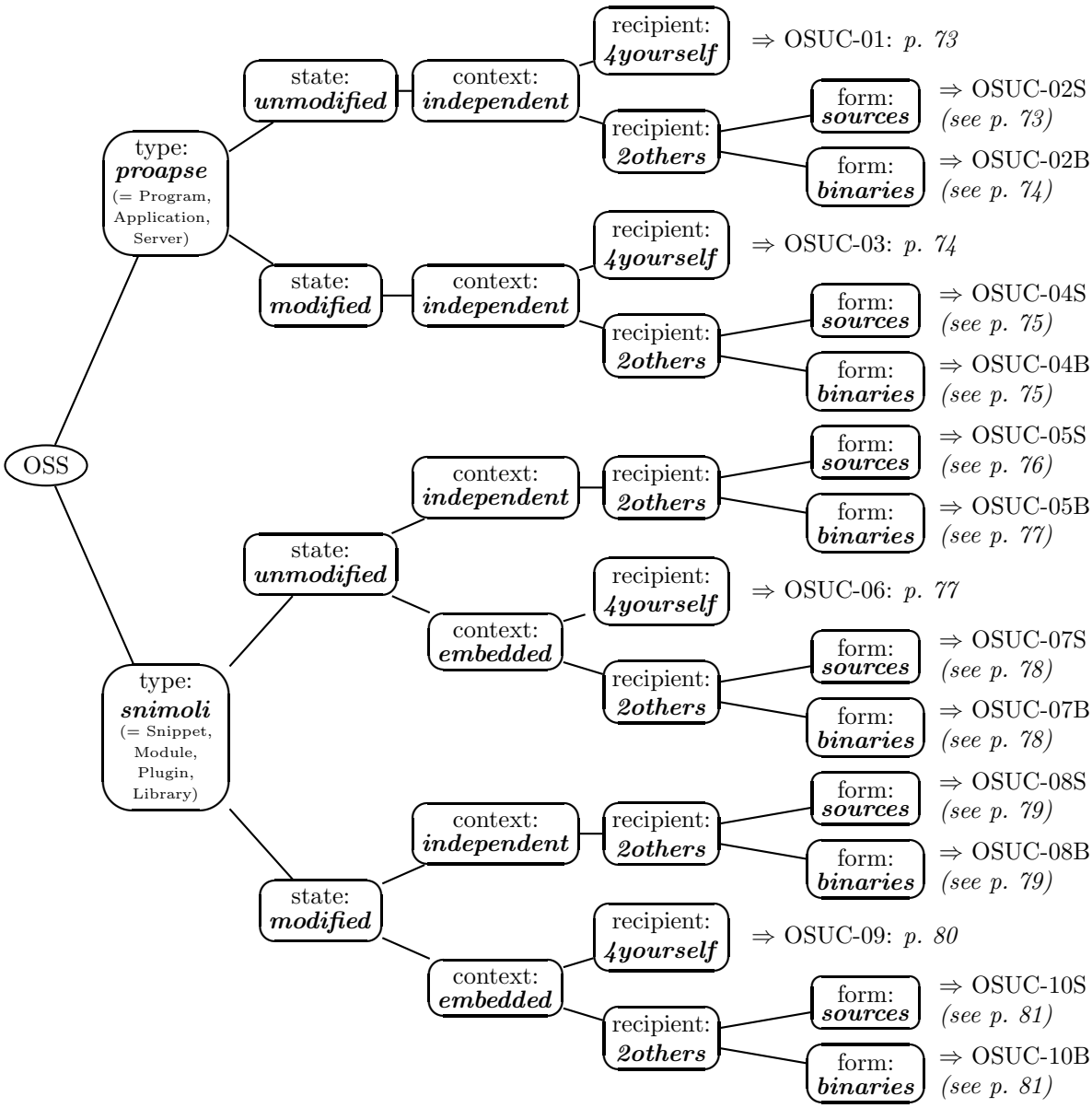
**Form:** Mostly it is up to you to decide whether you want to distribute only the binaries or whether you are intentionally going to distribute the sources (too). But in some cases, you have to respect some special conditions.[289].

[288] Please remember that – at least in Germany – there are opinions that even handing over software to another legal entity or department of the same company is also a kind of distribution. It is always safest to take the broadest possible meaning of distributing or handing over.

[289] For details concerning a necessary refinement of the open source use case taxonomy, please see → OSLiC, p. **??**

## 5.2 The taxonomic Open Source Use Case Finder

Now, after having gathered the necessary information, determine your specific open source use case by traversing the following tree and its corresponding branches:




## 5.3 The open source use cases and its to-do list references

On the following pages, each **O**pen **S**ource **U**se **C**ase is textually specified one more time and complemented by a list of page numbers. Each of these pages covers the license-specific to-do list whose items together offer a processable way

for acting according to the license under the circumstances of the described **O**pen **S**ource **U**se **C**ase.

**OSUC-01:** Only for yourself, you are going to use an unmodified open source program, application, or server – just as you received it. But you do not combine it with other components in the sense of software development (= *proapse, unmodified, independent, 4yourself*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 85 for the **ApL** *(= Apache License)*
- p. 95 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 104 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 109 for the **EPL** *(= Eclipse Public License)*
- p. 123 for the **EUPL** *(= European Union Public License)*
- p. 138 for the **GPL** *(= GNU General Public License)*
- p. 155 for the **LGPL** *(= GNU Lesser General Public License*
- p. 171 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 176 for the **MPL** *(= Mozilla Public License)*
- p. 190 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 201 for the **PHP** License

**OSUC-02S:** Just as you received it, you are going to distribute an unmodified open source program, application, or server to 3rd parties – in the form of sources. In this act of distribution, you do not combine this program, application, or server with other software components in the sense of software development (= *proapse, unmodified, independent, 2others, sources*) To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 85 for the **ApL** *(= Apache License)*
- p. 96 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 105 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 110 for the **EPL** *(= Eclipse Public License)*
- p. 124 for the **EUPL** *(= European Union Public License)*
- p. 138 for the **GPL** *(= GNU General Public License)*

- p. 156 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 177 for the **MPL** *(= Mozilla Public License)*
- p. 191 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 202 for the **PHP** License

**OSUC-02B:** Just as you received it, you are going to distribute an unmodified open source program, application, or server to 3rd parties – in the form of binaries. In this act of distribution, you do not combine this program, application, or server with other software components in the sense of software development (= *proapse, unmodified, independent, 2others, binaries*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 86 for the **ApL** *(= Apache License)*
- p. 96 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 105 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 110 for the **EPL** *(= Eclipse Public License)*
- p. 124 for the **EUPL** *(= European Union Public License)*
- p. 139 for the **GPL** *(= GNU General Public License)*
- p. 157 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 178 for the **MPL** *(= Mozilla Public License)*
- p. 191 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 202 for the **PHP** License

**OSUC-03:** Only for yourself, you are going to modify a received open source program, application, or server, before you will use it. But you do not combine it with other components in the sense of software development (= *proapse, modified, independent, 4yourself*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 85 for the **ApL** *(= Apache License)*
- p. 95 for the **BSD** License *(= Berkeley Software Distribution)*

- p. 104 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 109 for the **EPL** *(= Eclipse Public License)*
- p. 123 for the **EUPL** *(= European Union Public License)*
- p. 138 for the **GPL** *(= GNU General Public License)*
- p. 155 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 171 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 176 for the **MPL** *(= Mozilla Public License)*
- p. 190 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 201 for the **PHP** License

**OSUC-04S:** You are going to modify a received open source program, application, or server, before you will distribute it to 3rd parties – in the form of sources. But you do not combine this modified program, application, or server with other software components in the sense of software development (*= proapse, modified, independent, 2others, sources*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 87 for the **ApL** *(= Apache License)*
- p. 97 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 105 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 112 for the **EPL** *(= Eclipse Public License)*
- p. 128 for the **EUPL** *(= European Union Public License)*
- p. 143 for the **GPL** *(= GNU General Public License)*
- p. 158 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 179 for the **MPL** *(= Mozilla Public License)*
- p. 191 for the **MS-PL** *(= Microsoft Public License)*
- p. 198 for the **PGL** *(= Postgres License)*
- p. 203 for the **PHP** License

**OSUC-04B:** You are going to modify a received open source program, application, or server, before you will distribute it to 3rd parties – in the form of binaries. But you do not combine this modified program, application, or server with other software components in the sense of software development

(= *proapse, modified, independent, 2others, binaries*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 88 for the **ApL** *(= Apache License)*
- p. 98 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 106 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 113 for the **EPL** *(= Eclipse Public License)*
- p. 129 for the **EUPL** *(= European Union Public License)*
- p. 144 for the **GPL** *(= GNU General Public License)*
- p. 159 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 180 for the **MPL** *(= Mozilla Public License)*
- p. 192 for the **MS-PL** *(= Microsoft Public License)*
- p. 198 for the **PGL** *(= Postgres License)*
- p. 204 for the **PHP** License

**OSUC-05S:** Just as you received it, you are going to distribute an unmodified open source library, code snippet, module, or plugin to 3rd parties – in the form of sources. In this act of distribution, you do not combine this library, code snippet, module, or plugin with other software components in the sense of software development (= *snimoli, unmodified, independent, 2others, sources*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 85 for the **ApL** *(= Apache License)*
- p. 96 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 105 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 110 for the **EPL** *(= Eclipse Public License)*
- p. 124 for the **EUPL** *(= European Union Public License)*
- p. 138 for the **GPL** *(= GNU General Public License)*
- p. 156 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 177 for the **MPL** *(= Mozilla Public License)*
- p. 191 for the **MS-PL** *(= Microsoft Public License)*

- p. 197 for the **PGL** *(= Postgres License)*
- p. 202 for the **PHP** License

**OSUC-05B:** Just as you received it, you are going to distribute an unmodified open source library, code snippet, module, or plugin to 3rd parties – in the form of binaries. In this act of distribution, you do not combine this library, code snippet, module, or plugin with other software components in the sense of software development (= *snimoli, unmodified, independent, 2others, binaries*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 86 for the **ApL** *(= Apache License)*
- p. 96 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 105 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 110 for the **EPL** *(= Eclipse Public License)*
- p. 124 for the **EUPL** *(= European Union Public License)*
- p. 139 for the **GPL** *(= GNU General Public License)*
- p. 157 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 178 for the **MPL** *(= Mozilla Public License)*
- p. 191 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 202 for the **PHP** License

**OSUC-06:** Only for yourself and just as you received it, you are going to combine an unmodified open source library, code snippet, module, or plugin into a larger software unit as one of its parts. (= *snimoli, unmodified, embedded, 4yourself*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 85 for the **ApL** *(= Apache License)*
- p. 95 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 104 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 109 for the **EPL** *(= Eclipse Public License)*
- p. 123 for the **EUPL** *(= European Union Public License)*

- p. 138 for the **GPL** *(= GNU General Public License)*
- p. 155 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 171 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 176 for the **MPL** *(= Mozilla Public License)*
- p. 190 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 201 for the **PHP** License

**OSUC-07S:** Just as you received it and before you will distribute it to 3rd parties – in the form of sources and together with the larger software unit, you are going to combine and embed an unmodified open source library, code snippet, module, or plugin into a larger software unit in the sense of software development (= *snimoli, unmodified, embedded, 2others, sources*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 85 for the **ApL** *(= Apache License)*
- p. 96 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 105 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 110 for the **EPL** *(= Eclipse Public License)*
- p. 125 for the **EUPL** *(= European Union Public License)*
- p. 140 for the **GPL** *(= GNU General Public License)*
- p. 156 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 177 for the **MPL** *(= Mozilla Public License)*
- p. 191 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 202 for the **PHP** License

**OSUC-07B:** Just as you received it and before you will distribute it to 3rd parties – in the form of binaries and together with the larger software unit, you are going to combine and embed an unmodified open source library, code snippet, module, or plugin into a larger software unit in the sense of software development (= *snimoli, unmodified, embedded, 2others, binaries*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*

- p. 86 for the **ApL** *(= Apache License)*
- p. 96 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 105 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 110 for the **EPL** *(= Eclipse Public License)*
- p. 126 for the **EUPL** *(= European Union Public License)*
- p. 141 for the **GPL** *(= GNU General Public License)*
- p. 157 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 172 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 178 for the **MPL** *(= Mozilla Public License)*
- p. 191 for the **MS-PL** *(= Microsoft Public License)*
- p. 197 for the **PGL** *(= Postgres License)*
- p. 202 for the **PHP** License

**OSUC-08S:** Before you will distribute it to 3rd parties in the form of sources, you are going to modify an open source library, code snippet, module, or plugin. But you do not combine it with other software components in the sense of software development (= *snimoli, modified, independent, 2others, sources*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 89 for the **ApL** *(= Apache License)*
- p. 99 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 106 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 114 for the **EPL** *(= Eclipse Public License)*
- p. 130 for the **EUPL** *(= European Union Public License)*
- p. 145 for the **GPL** *(= GNU General Public License)*
- p. 161 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 173 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 182 for the **MPL** *(= Mozilla Public License)*
- p. 193 for the **MS-PL** *(= Microsoft Public License)*
- p. 199 for the **PGL** *(= Postgres License)*
- p. 204 for the **PHP** License

**OSUC-08B:** Before you will distribute it to 3rd parties in the form of binaries, you are going to modify an open source library, code snippet, module, or

plugin. But you do not combine it with other software components in the sense of software development (= *snimoli, modified, independent, 2others*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 90 for the **ApL** *(= Apache License)*
- p. 99 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 107 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 115 for the **EPL** *(= Eclipse Public License)*
- p. 131 for the **EUPL** *(= European Union Public License)*
- p. 147 for the **GPL** *(= GNU General Public License)*
- p. 162 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 173 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 183 for the **MPL** *(= Mozilla Public License)*
- p. 193 for the **MS-PL** *(= Microsoft Public License)*
- p. 199 for the **PGL** *(= Postgres License)*
- p. 205 for the **PHP** License

**OSUC-09:** Only for yourself, you are going to modify an open source library, code snippet, module, or plugin, and you will combine it – in the sense of software development – into a larger software unit as one of its parts. (= *snimoli, modified, embedded, 4yourself*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 85 for the **ApL** *(= Apache License)*
- p. 95 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 104 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 109 for the **EPL** *(= Eclipse Public License)*
- p. 123 for the **EUPL** *(= European Union Public License)*
- p. 138 for the **GPL** *(= GNU General Public License)*
- p. 155 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 171 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 176 for the **MPL** *(= Mozilla Public License)*
- p. 190 for the **MS-PL** *(= Microsoft Public License)*

- p. 197 for the **PGL** *(= Postgres License)*
- p. 201 for the **PHP** License

**OSUC-10S:** Before you will distribute it to 3rd parties in the form of sources, you are going to modify an open source library, code snippet, module, or plugin, which you combine with other software components in the sense of software development (= *snimoli, modified, independent, 2others, sources*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 91 for the **ApL** *(= Apache License)*
- p. 100 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 107 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 116 for the **EPL** *(= Eclipse Public License)*
- p. 132 for the **EUPL** *(= European Union Public License)*
- p. 148 for the **GPL** *(= GNU General Public License)*
- p. 164 for the **LGPL** *(= GNU Lesser General Public License)*
- p. 174 for the **MIT** License *(= Massachusetts Institute of Technology)*
- p. 184 for the **MPL** *(= Mozilla Public License)*
- p. 194 for the **MS-PL** *(= Microsoft Public License)*
- p. 199 for the **PGL** *(= Postgres License)*
- p. 206 for the **PHP** License

**OSUC-10B:** Before you will distribute it to 3rd parties in the form of binaries, you are going to modify an open source library, code snippet, module, or plugin, which you combine with other software components in the sense of software development (= *snimoli, modified, independent, 2others, binaries*). To see the *specific, license fulfilling to-do lists* jump to the following pages:

- p. 84 for the **AGPL** *(= GNU Affero General Public License)*
- p. 92 for the **ApL** *(= Apache License)*
- p. 101 for the **BSD** License *(= Berkeley Software Distribution)*
- p. 107 for the **CDDL** *(= Common Develop and Distribution License)*
- p. 118 for the **EPL** *(= Eclipse Public License)*
- p. 133 for the **EUPL** *(= European Union Public License)*
- p. 149 for the **GPL** *(= GNU General Public License)*
- p. 165 for the **LGPL** *(= GNU Lesser General Public License)*

# 6 Open Source License Compliance: To-Do Lists

*With respect to the defined open source use cases, this chapter lists what one has to do for acting in accordance with the specific open source licenses.*

## 6.1 Some general remarks on 'giving' someone a file

This chapter has to be started with some general points which are relevant for many of the to-do lists. So that the same points are not repeated too often, we will start with these general remarks and refer to them throughout the chapter

- Sometimes when delivering a binary package containing open source software, the medium doesn't allow the recipient to view all files contained in that package. For example, a lot of mobile devices don't give the user access to the file system. But open source licenses often require 'to give' someone copies of text files, such as the license text, copyright notes, or specific notice file. The safe interpretation of 'giving someone a text' is that the receiver must be able to read it[290]. Thus, on systems which offer a file browser and a suitable reader, it is sufficient, to put these file onto the files system. On the other systems, you *must* present the content of the files through the UI of your application – for example in a specific copyright screen[291]. The OSLiC does not want to refine the taxonomies down to the level of operating systems, so it is up to the user to keep this in mind when reading the to-do lists.
- Sometimes a product which uses and distributes open source software tries to fulfill the requirement 'to give the recipients the license etc.' by presenting links to general versions of these licensing files hosted somewhere on the internet. But be aware: Although it is a good tradition – especially if you link to the homepages of the projects for being totally transparent – it is not sufficient to offer only the links. If you are required by the open source licenses to handover something to your users, *you* must do it. It is not safe to delegate the task to anyone hoping that they will offer the files all the time your product is being distributed[292]. Even if it would be safe

[290] To give someone anything they can't touch, feel or see is like not giving him the object ;-)

[291] Additionally, in the open source community, it is a good tradition, to present these reference data voluntarily.

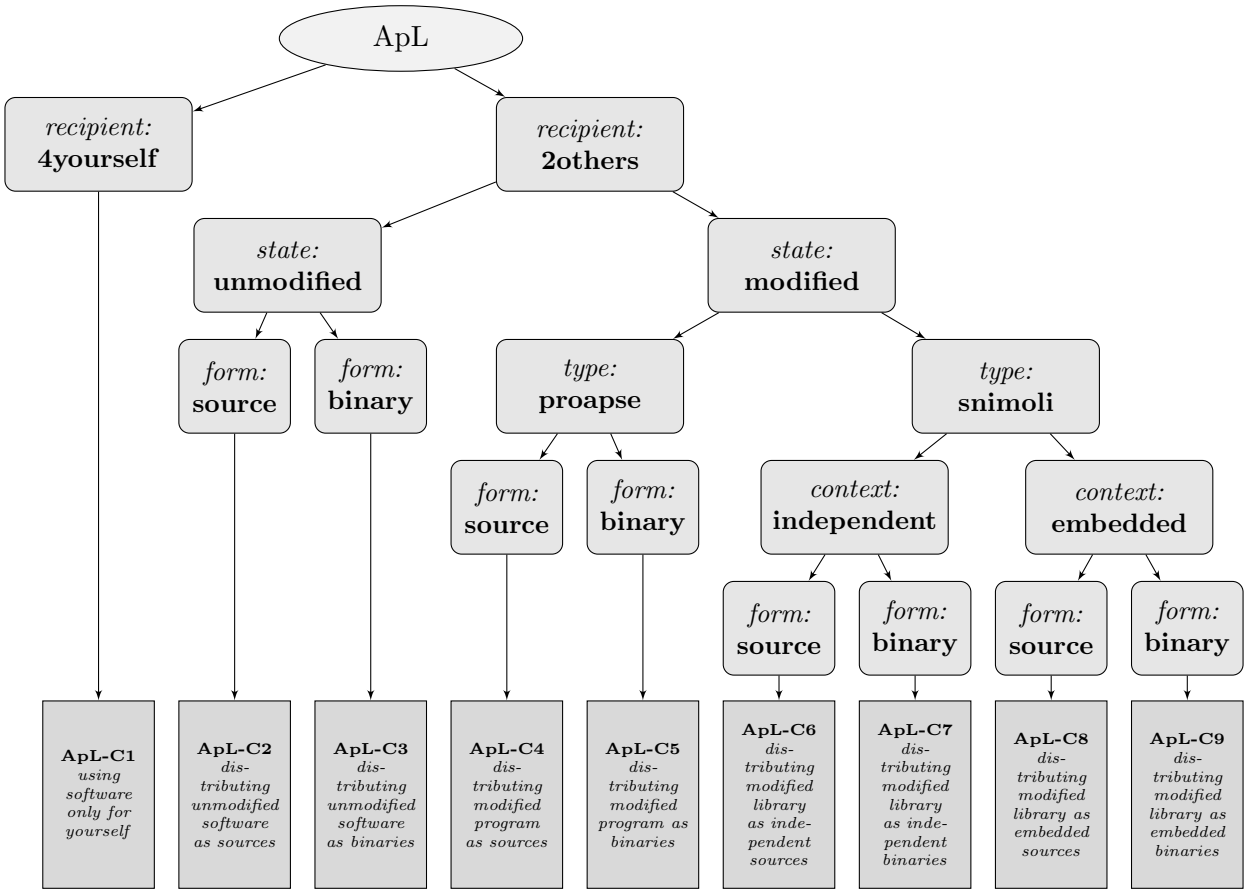
[292] Moreover, the advantage of doing the job oneself is that one has not to struggle with uncommunicated implicite modifications of the link targets.

to assume that the link will remain valid forever, the point is: you have to fulfill the license, no one else.

## 6.2 AGPL Licensed Software in the usage context of . . . [tbd]

## 6.3 Apache licensed software

Today, the current release of the Apache open source license is version 2.0, elder versions are depricated[293]. Because it focusses primarily on the "redistribution"[294], the following simplified Apache specific open source use case finder[295] can be used:




293) For details → OSLiC, pp. 26
294) cf. *Open Source Initiative*: APL-2.0, 2004, wp. §4.
295) For details of the general OSUC finder → OSLiC, pp. 66 and 69

### 6.3.1 ApL-C1: Using the software only for yourself

**means** that you are going to use a received Apache licensed software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[296]

**requires** no tasks in order to fulfill the conditions of the Apache 2.0 license with respect to this use case:

- You are allowed to use any kind of Apache software in any sense and in any context without being obliged to do anything as long as you do not give the software to 3rd parties.

**prohibits** ...

- to promote any of your services – based on the this software – by trademarks, service marks, or product names linked to the software except as required for unpartially describing the used software file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.2 ApL-C2: Passing the unmodified software as source code

**means** that you are going to distribute an unmodified version of the received Apache software to 3rd parties – in the form of source code files or as a source code package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or as an embedded unit.

**covers** OSUC-02S, OSUC-05S, OSUC-07S[297]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the software package, add it[298].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that a *notice text file*[299] is retained in your package in the form you have received it.

[296] For details → OSLiC, pp. 73 - 80

[297] For details → OSLiC, pp. 73 - 78

[298] For implementing the handover of files correctly → OSLiC, p. 83

[299] The Apache license seems purposely to be a bit ambiguous: it uses the term ""Notice" text file". In its strict sense, the term refers to a file named 'NOTICE.[txt|pdf|...]'. In a weaker

- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*, a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.3 ApL-C3: Passing the unmodified software as binaries

**means** that you are going to distribute an unmodified version of the received Apache software to 3rd parties – in the form of binary files or as a binary package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02B, OSUC-05B, OSUC-07B[300]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the binary package, add it[301].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.
- [**mandatory:**] Ensure that the *notice text file* is retained or integrated into your binary package in the form you have initially received it.
- [**mandatory:**] Ensure that the *notice text file* is also reproduced if and whereever such third-party notices normally appear – especially, if you are distributing an unmodified Apache licensed library as embedded component of your own work which displays its own copyright notice.

sense, it may denote any (text) file containing (licensing) notices. For being sure to act according to this requirement you should also read this term in the broader sense if there is no text file named 'NOTICE'

[300] For details → OSLiC, pp. 74 - 78

[301] For implementing the handover of files correctly → OSLiC, p. 83

- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*, a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license – especially as subsection of your own copyright notice.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.4 ApL-C4: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received Apache licensed program, application, or server (proapse) to 3rd parties – in the form of source code files or as a source code package.

**covers** OSUC-04S[302]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the software package, add it[303].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the *notice text file* contains at least all the information of that *notice text file* you have received.
- [**mandatory:**] Ensure that the *notice text file* is also reproduced if and whereever such third-party notices normally appear. If the program already displays a copyright dialog, update it in an appropriate manner.
- [**mandatory:**] Inside of the source code, mark all your modifications thoroughly. Generate a *notice text file*, if it still does not exist. *Add* a description of your modifications into the *notice text file*.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*,

[302] For details → OSLiC, pp. 75
[303] For implementing the handover of files correctly → OSLiC, p. 83

a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.5 ApL-C5: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received Apache licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[304]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the binary package, add it[305].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.
- [**mandatory:**] Ensure that the *notice text file* contains at least all the information of that *notice text file* you have received. If it still does not exist, create it. *Expand* the *notice text file* by a description of your modifications.
- [**mandatory:**] Ensure that the *notice text file* is also reproduced if and whereever such third-party notices normally appear. If the program already displays a copyright dialog, update it in an appropriate manner.
- [**voluntary:**] Even if you do not want to distribute your modified source code, mark all your modifications thoroughly.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*,

[304] For details → OSLiC, pp. 75

[305] For implementing the handover of files correctly → OSLiC, p. 83

a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.6 ApL-C6: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received Apache licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[306]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the software package, add it[307].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the *notice text file* contains at least all the information of that *notice text file* you have received.
- [**voluntary:**] Inside of the source code, mark all your modifications thoroughly. Generate a *notice text file*, if it still does not exist. *Expand* the *notice text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*, a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license.

**prohibits** ...

[306] For details → OSLiC, pp. 79
[307] For implementing the handover of files correctly → OSLiC, p. 83

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.7 ApL-C7: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received Apache licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[308]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the binary package, add it[309].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.
- [**mandatory:**] Ensure that the *notice text file* contains at least all the information of that *notice text file* you have received. If it still does not exist, create it. *Expand* the *notice text file* by a description of your modifications.
- [**voluntary:**] Even if you do not want to distribute your modified source code, mark all your modifications thoroughly.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*, a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this

[308] For details → OSLiC, pp. 79

[309] For implementing the handover of files correctly → OSLiC, p. 83

Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.

- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.8 ApL-C8: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received Apache licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[310]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the software package, add it[311].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the *notice text file* contains at least all the information of that *notice text file* you have received.
- [**mandatory:**] Ensure that the *notice text file* is also reproduced if and whereever such third-party notices normally appear. If your overarching program displays its own copyright dialog, insert this information there.
- [**mandatory:**] Inside of the library[312] source code, mark all your modifications thoroughly. Generate a *notice text file*, if it still does not exist. *Expand* the *notice text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*, a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license.
- [**voluntary:**] Arrange your source code distribution so that the integrated Apache license and the *notice text file* clearly refer only to the

[310] For details → OSLiC, pp. 81
[311] For implementing the handover of files correctly → OSLiC, p. 83
[312] or snippet, or module, or plugin

embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the embedded components like libraries, modules, snippets, or plugins in specific directory which contains also all additional licensing elements.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.9 ApL-C9: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received Apache licensed code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[313]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Give the recipient a copy of the Apache 2.0 license. If it is not already part of the binary package, add it[314].
- [**mandatory:**] Ensure that the licensing elements – esp. the specific copyright notice of the original author(s) – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.
- [**mandatory:**] Ensure that the *notice text file* contains at least all the information of that *notice text file* you have received. If it still does not exist, create it. *Expand* the *notice text file* by a description of your modifications.
- [**mandatory:**] Ensure that the *notice text file* is also reproduced if and whereever such third-party notices normally appear. If your overarching program displays its own copyright dialog, insert this information there.

[313] For details → OSLiC, pp. 81
[314] For implementing the handover of files correctly → OSLiC, p. 83

- [**voluntary:**] Even if you do not want to distribute your modified source code, mark all your modifications of the embedded libary[315] thoroughly.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the *notice text file*, a hint to the software name, a link to its homepage, and a link to the Apache 2.0 license – especially as subsection of your own copyright notice.
- [**voluntary:**] Arrange your binary distribution so that the integrated Apache license and the *notice text file* clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the libraries, modules, snippet, or plugins in specific directories which contain also all licensing elements.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trademarks, service marks, or product names linked to this Apache software, except as required for unpartially describing the used software and for reproducing the notice text file.
- to institute any patent litigation against anyone alleging that the software constitutes patent infringement.

### 6.3.10 Discussions and Explanations

- On the one hand, the Apache 2.0 license does not permit "[...] to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file"[316]. On the other hand, this license alerts that all the patent licenses – granted to those who "[...] institute a patent litigation" – will terminate automatically[317]. Hence, the OSLiC generally (ApL-C1 - ApL-C9) interdicts to promote products or services by these elements and to legally fight against patents linked to the software.
- The ApL also requires to "[...] give any other recipients of the Work or Derivative Works a copy of this License"[318]. Therefore, all *2others* use cases contain the respective mandatory condition (ApL-C2 - ApL-C9).

[315] or snippet, or module, or plugin
[316] cf. *Open Source Initiative*: APL-2.0, 2004, wp. §6.
[317] cf. id., l.c., wp. §3.
[318] cf. id., l.c., wp. §4.1.

- Additionally, the ApL requires, that modifications must be marked[319]. Thus, in all cases of passing the modified software in the form of source code the OSLiC requires to mark the modifications and to integrate a hint into the notice file – while in all the cases of passing the modified software in the form of binaries it inserts only a voluntary condition (ApL-C4 - ApL-C9).
- Furthermore, the ApL requires that one must "[...] retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work" So, the OSLIC requires in all contexts (ApL-C1 - ApL-C9) that the licensing elements are retained in the form you have received them[320].
- Finally, the ApL requires that the received "NOTICE text file" must be integrated as readable copy to each package distributed in the form of source code, or – in case of binary distibutions – must be displayed "[...] if and wherever such third-party notices normally appear"[321]. Thus, the OSLiC requires mandatorily that all source code distributions must include the notice text file (ApL-C2, ApL-C4, ApL-C6, ApL-C8) and that all distributions of binary applications which normally show such a copyrigth screen must integrate the content of the notice file into this screen (ApL-C5, ApL9). For libraries distributed in the form of binaries it is assumed that they normally do not contain such copyright dialogs (ApL-C7)

## 6.4 BSD licensed software

As an approved open source license, the BSD license exists in two versions[322]. The latest release is the *BSD 2-Clause license*[323], the elder release is the *BSD 3-Clause license*[324]. The very little differences between the two versions have to be respected exactly. Nevertheless, we could integrate the requirements into one to-do list per use case.

319) cf. *Open Source Initiative*: APL-2.0, 2004, wp. §4.2.

320) This might confuse some readers: Yes, even if you distribute a modified version in the form of binaries you must fulfill this condition. Moreover, you must also hand the license over to your receipient. But, nevertheless, you are not obliged to publish the modified source code, too. (→ OSLiC, p. 26)

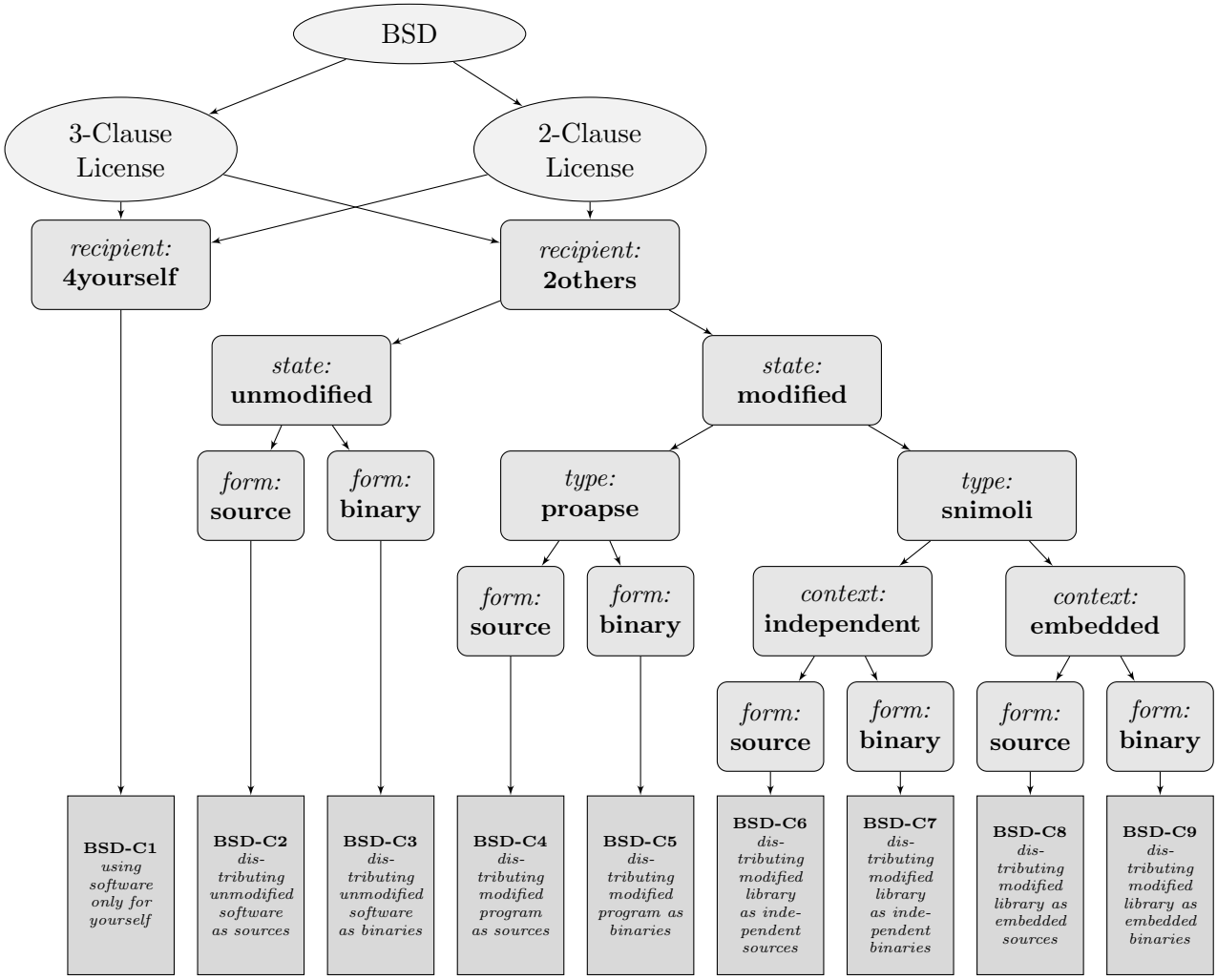
321) cf. id., l.c., wp. §4.4.

322) Following the OSI, there is another 'ancient' BSD license – containing a fourth clause known as advertising clause – which "(...) officially was rescinded by the Director of the Office of Technology Licensing of the University of California on July 22nd, 1999". Cf. *Open Source Initiative*: The BSD 3-Clause License, 2012, wp. Because of that cancellation you can simply act according the *BSD 3-Clause license* if you have to fulfill the eldest BSD license.

323) cf. *Open Source Initiative*: The BSD 2-Clause License, 2012, wp.

324) cf. *Open Source Initiative*: The BSD 3-Clause License, 2012, wp.

Explicitly, all BSD open source licenses 'only' focus on the (re-)distribution *open source use cases* which we have specified by our token *2others*. Conditions for the other use cases specified by the token *4yourself* can be derived[325]. Additionally the BSD licenses consider the form of the distribution, esp. whether the work is distributed as a (set of) source code file(s) or as a (set of) the binary file(s). Use the following tree to find the BSD license fulfilling to-do lists.




### 6.4.1 BSD-C1: Using the software only for yourself

**means** that you are going to use a received BSD software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[326]

[325] For details of the *open source use case tokens* see p. 66. For details of the *open source use cases* based on these token see p. 69

[326] For details → OSLiC, pp. 73 - 80

**requires** no tasks in order to fulfill the conditions of the BSD license with respect to this use case:

- You are allowed to use any kind of BSD software in any sense and in any context without any obligations as long as you do not give the software to 3rd parties.

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work[327].

### 6.4.2 BSD-C2: Passing the unmodified software as source code

**means** that you are going to distribute an unmodified version of the received BSD software to 3rd parties – in the form of source code files or as a source code package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02S, OSUC-05S, OSUC-07S[328]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the BSD license text, the specific copyright notice of the original author(s), and the BSD disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the BSD conditions, and the BSD disclaimer.

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

### 6.4.3 BSD-C3: Passing the unmodified software as binary

**means** that you are going to distribute an unmodified version of the BSD received software to 3rd parties – in the form of binary files or as a binary package. In

[327] which may be - for example - an internet service based on this BSD software used in your own data center

[328] For details → OSLiC, pp. 73 - 78

this case it it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02B, OSUC-05B, OSUC-07B[329]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that your distribution contains the original copyright notice, the BSD license, and the BSD disclaimer in the form you have received them. If you compile the binary file on the base of the source code package and if this compilation does not also generate and integrate the licensing files then create the copyright notice, the BSD conditions, and the BSD disclaimer according to the form of the source code package and insert these files into your distribution manually[330].
- [**mandatory:**] Ensure that the documentation of your distribution and/or your additional material also contain the author specific copyright notice, the BSD conditions, and the BSD disclaimer.

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

### 6.4.4 BSD-C4: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received BSD program, application, or server (proapse) to 3rd parties in the form of source code files or as a source code package.

**covers** OSUC-04S[331]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the BSD license text, the specific copyright notice of the original author(s), and the BSD disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the BSD conditions, and the BSD disclaimer.

[329] For details → OSLiC, pp. 74 - 78
[330] For implementing the handover of files correctly → OSLiC, p. 83
[331] For details → OSLiC, pp. 75

- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the program is licensed under the BSD license. Because you are already modifying the program you can also add such a hint if the presented original copyright notice lacks such a statement.

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

### 6.4.5 BSD-C5: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received BSD program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[332]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that your distribution contains the original copyright notice, the BSD license, and the BSD disclaimer in the form you have received them. If you compile the binary file on the base of the source code package and if this compilation does not also generate and integrate the licensing files then create the copyright notice, the BSD conditions, and the BSD disclaimer according to the form of the source code package and insert these files into your distribution manually[333].
- [**mandatory:**] Ensure that the documentation of your distribution and/or your additional material also contain the author specific copyright notice, the BSD conditions, and the BSD disclaimer.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the program is licensed under the BSD license. Because you are already modifying the program you can also add such a hint if the presented original copyright notice lacks such a statement.

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

[332] For details → OSLiC, pp. **??**

[333] For implementing the handover of files correctly → OSLiC, p. 83

### 6.4.6 BSD-C6: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received BSD code snippet, module, library, or plugin (snimoli) to 3rd parties in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[334]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the BSD license text, the specific copyright notice of the original author(s), and the BSD disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the BSD conditions, and the BSD disclaimer.

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

### 6.4.7 BSD-C7: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received BSD code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[335]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that your distribution contains the original copyright notice, the BSD license, and the BSD disclaimer in the form you have received them. If you compile the binary file on the base of the source code package and if this compilation does not also generate and integrate the licensing files, then create the copyright notice, the BSD conditions, and the BSD disclaimer according to the form of the source code package and insert these files into your distribution manually[336].

[334] For details → OSLiC, pp. 79
[335] For details → OSLiC, pp. 79
[336] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Ensure that the documentation of your distribution and/or your additional material also contain the author specific copyright notice, the BSD conditions, and the BSD disclaimer.

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

### 6.4.8 BSD-C8: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received BSD code snippet, module, library, or plugin (snimoli) to 3rd parties in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[337]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the BSD license text, the specific copyright notice of the original author(s), and the BSD disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the BSD conditions, and the BSD disclaimer.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that it contains components licensed under the BSD license. Because you are embedding this snimoli into a larger software unit, you are developing this larger unit. Hence, you can also expand the copyright notice of this larger unit by such a hint to its BSD components.
- [**voluntary:**] Arrange your source code distribution so that the licensing elements – esp. the BSD license text, the specific copyright notice of the original author(s), and the BSD disclaimer – clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the embedded components like libraries, modules, snippets, or plugins in specific directory which contains also all additional licensing elements.

[337] For details → OSLiC, pp. 81

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

### 6.4.9 BSD-C9: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received BSD code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[338]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that your distribution contains the original copyright notice, the BSD license, and the BSD disclaimer in the form you have received them. If you compile the binary file on the base of the source code package and if this compilation does not also generate and integrate the licensing files, then create the copyright notice, the BSD conditions, and the BSD disclaimer according to the form of the source code package and insert these files into your distribution manually[339].
- [**mandatory:**] Ensure that the documentation of your distribution and/or your additional material also contain the author specific copyright notice, the BSD conditions, and the BSD disclaimer.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that it contains components licensed under the BSD license. Because you are embedding this snimoli into a larger software unit, you are developing this larger unit. Hence, you can also expand the copyright notice of this larger unit by such a hint to its BSD components.
- [**voluntary:**] Arrange your binary distribution so that the licensing elements – esp. the BSD license text, the specific copyright notice of the original author(s), and the BSD disclaimer – clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the libraries, modules, snippet, or plugins in specific directories which contain also all licensing elements.

[338] For details → OSLiC, pp. 81

[339] For implementing the handover of files correctly → OSLiC, p. 83

**prohibits** nothing explicitly if you are using the *BSD 2 Clause License*. But the *BSD 3 Clause License* explicitly prohibits to use the name of the licensing organization or the names of the licensing contributors to promote your own work.

### 6.4.10 Discussions and Explanations

The *BSD 2-Clause license* has a simple structure: In the beginning, it generally "(permits) [the] redistribution and [the] use in source and binary forms, with or without modification, [...]", if one fulfills the two rules of the license[340]. The first rule concerns the (re)distribution in the form of source code, the second the (re)distribution of binary packages. Here are some explanations why we translated the rules into which sets of executable tasks:

- For the "redistribution of source code" the license requires, that the package must " [...] retain the above copyright notice, this list of conditions and the following disclaimer"[341]. Hence, you are not allowed, to modify any of the copyright notes which are already embedded in the received (source) files. And from a logical point of view, there must exist an explicit or implicit assertion that the software is licensed under the *BSD 2-Clause license*[342]. This is often implemented by simply adding a copy of the license into the package. Hence, you are furthermore not allowed to modify these files or corresponding text snippets. For our purposes, we translated the bans into the following executable task:

  > *Ensure that the licensing elements – esp. the BSD license text, the specific copyright notice of the original author(s), and the BSD disclaimer – are retained in your package in the form you have received them.*

- For the redistribution in the form of binary files, the license requires, that the licensing elements must be "[...] (reproduced) in the documentation and/or other materials provided with the distribution"[343]. Hence, this is not required as a necessary condition for the (re)distribution as source code

[340] cf. *Open Source Initiative*: The BSD 2-Clause License, 2012, wp.
[341] cf. id., ibid.
[342] The BSD license requires that a re-distributed software package must contain the (package specific) copyright notice, the (license specific) conditions and the BSD disclaimer. (cf. id., l.c., wp) You might ask what you should do, if these elements are missed in the package you received. If so, the package you received had not been licensed adequately. Hence, you do not know reliably whether you have received it under a BSD license. In other words: If you have received a BSD licensed software package, it must contain sufficient license fulfilling elements, or it is not a BSD licensed software.
[343] cf. id., l.c., wp.

package. But nevertheless, even for a distribution in the form of source code, it is often possible to fulfill this rule too – e.g. if you offer your own download site for source code packages. In such cases, it is a sign of respect to mention the licensing not only inside of the packages, but also in the text of your site. Because of that, we added the following voluntary task for all BSD open source use cases which deal with the redistribution in the form of source code:

> *Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the BSD conditions, and the BSD disclaimer.*

- Naturally, because the reproduction of the licensing elements "in the documentation and/or other materials provided with the distribution" is explicitly required for the "redistribution in binary form"[344], we had to rewrite the facultative task for a distribution in the form of source code as a mandatory task for all BSD open source use cases which deals with the redistribution in binary form:

  > *Ensure that the documentation of your distribution and/or your additional material also contains the author specific copyright notice, the BSD conditions, and the BSD disclaimer.*

- In case of (re)distributing the program in the form of binary files, it is sometimes not enough, to pass the licensing elements as one has received them. If you compile the binary package from the source code, it is not necessarily true, that the licensing elements are also automatically generated and embedded into the 'binary package'. But nevertheless, you have to add the copyright notice, the conditions and the disclaimer to this package for acting according to the BSD license. Therefore we chose the following form of an executable, license fulfilling task for all binary oriented distributions:

  > *Ensure that your distribution contains the original copyright notice, the BSD license, and the BSD disclaimer in the form you have received them. If you compile the binary file on the base of the source code package and if this compilation does not also generate and integrate the licensing files, then create the copyright notice the BSD conditions, and the BSD disclaimer according to the form of the source code package and insert these files into your distribution manually.*

- Finally, we wished to insert a hint to the general (open source) tradition, to mention the used open source software and their licenses as a remark of the 'copyright widget' of an application. This is not required by the

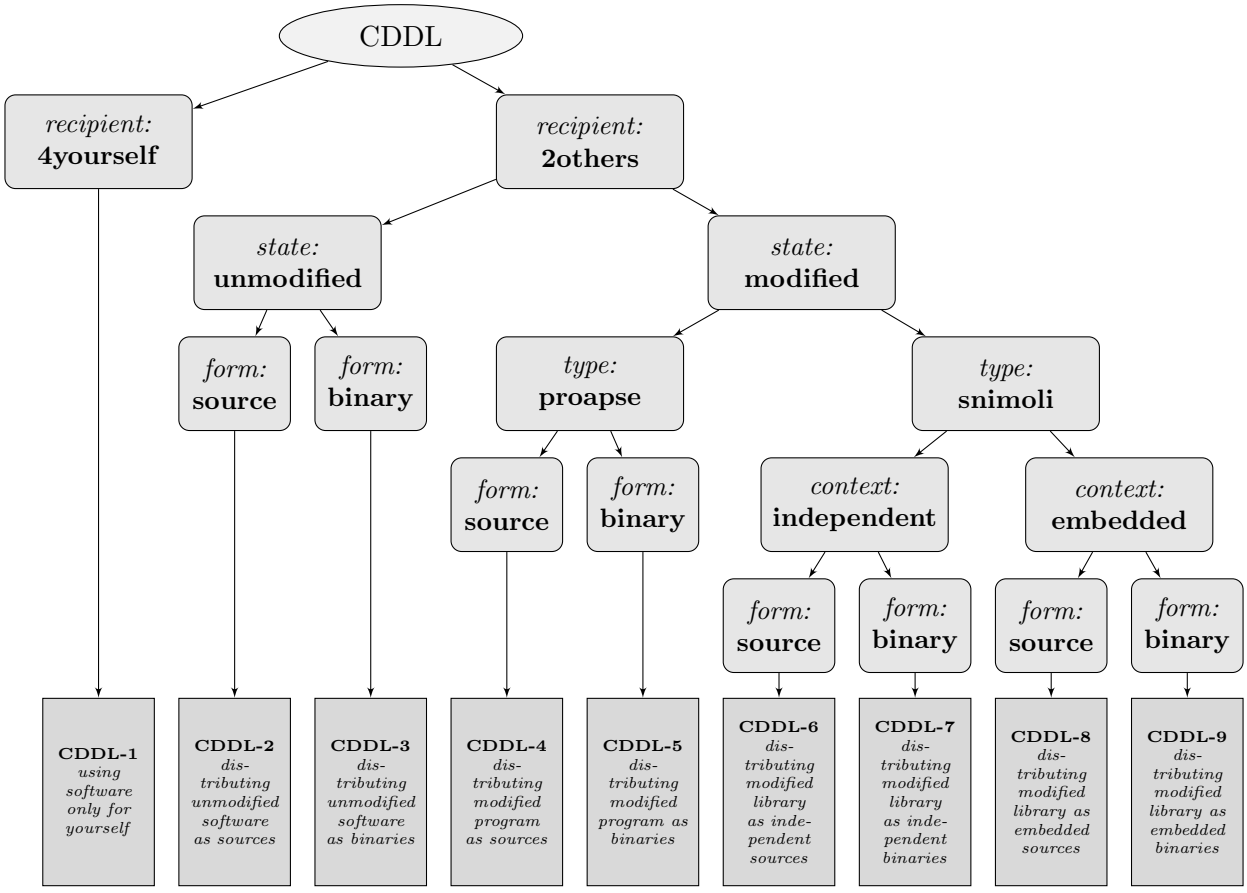
[344] cf. *Open Source Initiative*: The BSD 2-Clause License, 2012, wp.

BSD license. But it is a general, good tradition. Naturally, because of the freedom to use and modify open source software and to redistribute a modified version of it, you are also allowed to insert such references, even if they are missing. Therefore we added a third voluntary license tradition fulfilling task for all relevant open source use cases.

## 6.5 CDDL licensed software [tbd]

Also, [...]

Thus, for finding the relevant, simply processable task lists, also the following CDDL specific open source use case structure[345] can be used:




### 6.5.1 CDDL-1: Using the software only for yourself

**means** that you are going to use a received CDDL licensed software only for yourself and that you do not hand it over to any 3rd party in any sense.

[345] For details of the general OSUC finder → OSLiC, pp. 66 and 69

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[346]

**requires** ...

**prohibits** ...

### 6.5.2 CDDL-2: Passing the unmodified software as source code

**means** that you are going to distribute an unmodified version of the received CDDL software to 3rd parties - in the form of source code files or as a source code package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit

**covers** OSUC-02S, OSUC-05S, OSUC-07S[347]

**requires** the following tasks in order to fulfill the license conditions:

- ...

**prohibits** ...

- ...

### 6.5.3 CDDL-3: Passing the unmodified software as binaries

**means** that you are going to distribute an unmodified version of the received CDDL software to 3rd parties – in the form of binary files or as a binary package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02B, OSUC-05B, OSUC-07B[348]

**requires** the following tasks in order to fulfill the license conditions:

- ...

**prohibits** ...

- ...

### 6.5.4 CDDL-4: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received CDDL licensed program, application, or server (proapse) to 3rd parties – in the

[346] For details → OSLiC, pp. 73 - 80
[347] For details → OSLiC, pp. 73 - 78
[348] For details → OSLiC, pp. 74 - 78

form of source code files or a source code package.

**covers** OSUC-04S[349]

**requires** the following tasks in order to fulfill the license conditions:

- ...

**prohibits** ...

- ...

### 6.5.5 CDDL-5: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received CDDL licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[350]

**requires** the following tasks in order to fulfill the license conditions:

- ...

**prohibits** ...

- ...

### 6.5.6 CDDL-6: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received CDDL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[351]

**requires** the following tasks in order to fulfill the license conditions:

- ...

**prohibits** ...

- ...

[349] For details → OSLiC, pp. 75
[350] For details → OSLiC, pp. 75
[351] For details → OSLiC, pp. 79

### 6.5.7 CDDL-7: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received CDDL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[352]

**requires** the following tasks in order to fulfill the license conditions:

- ...

**prohibits** ...

- ...

### 6.5.8 CDDL-8: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received CDDL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[353]

**requires** the following tasks in order to fulfill the license conditions:

- ...

**prohibits** ...

- ...

### 6.5.9 CDDL-9: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received CDDL licensed code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[354]

**requires** the following tasks in order to fulfill the license conditions:

[352] For details → OSLiC, pp. 79
[353] For details → OSLiC, pp. 81
[354] For details → OSLiC, pp. 81

- ...

**prohibits** ...

- ...

### 6.5.10 Discussions and Explanations

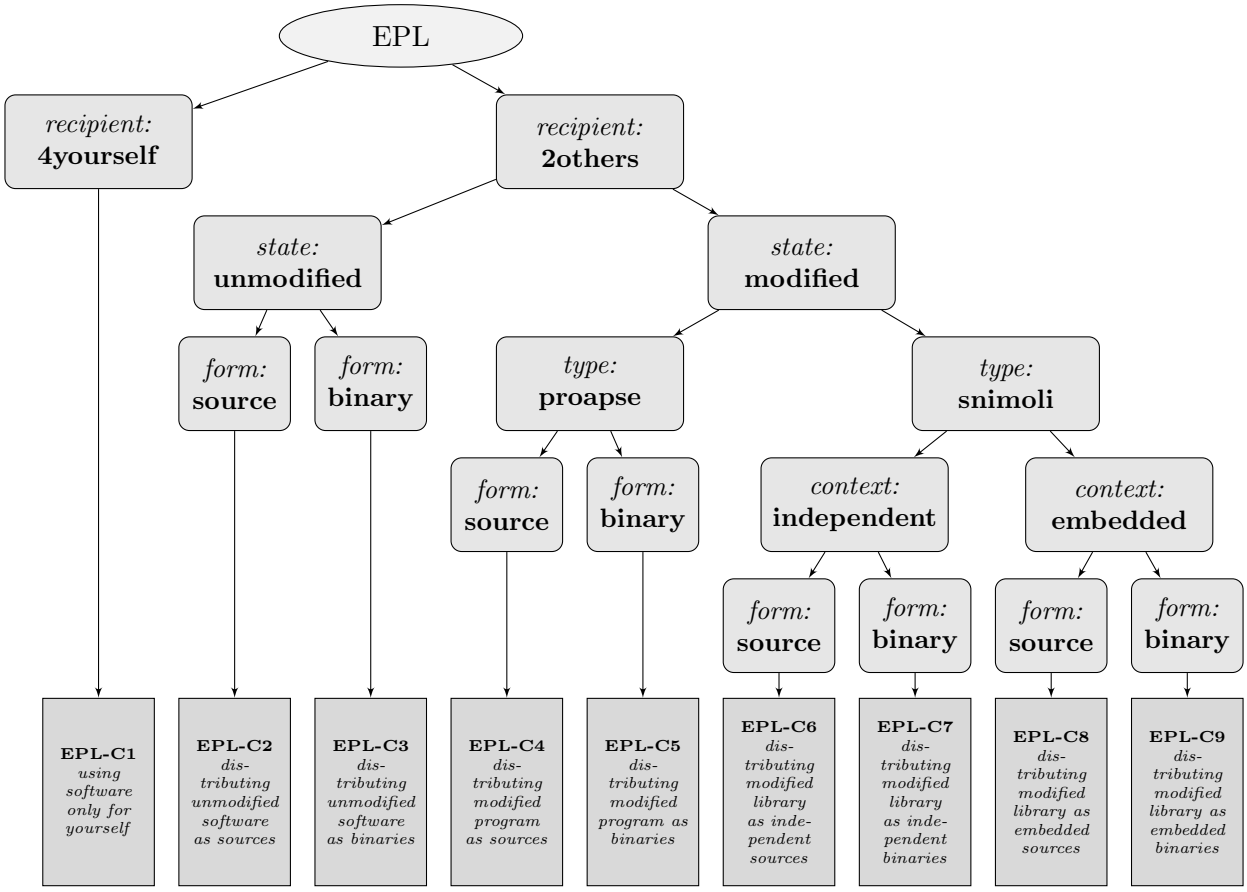
The CDDL offers ... which contains nearly all requirements[355]. Only for some

-

## 6.6 EPL licensed software

The Eclipse Public License clearly distinguishes the distribution in the form of source code from that in the form of binaries: First, it allows to "distribute" Eclipse licensed programs "in source code and in object code"[356]. Then it specifies under which conditions one may distribute the program as a set of binaries[357]. One of these conditions is – roughly spoken – that the distributor makes the sources available too[358]. The other conditions refer to the distribution in general – no matter what form or state is used[359]. So, taken as whole, the EPL mainly focusses on the distribution of software. Thus, for finding the relevant, simply processable task lists, also the following EPL specific open source use case structure[360] can be used:

[355] cf. *Open Source Initiative*: The CDDL-1.0, 2004, wp. §3.
[356] cf. *Open Source Initiative*: EPL-1.0, 2005, wp. §3.
[357] cf. id., l.c., wp. §3 top area.
[358] cf. id., l.c., wp. §3 mid area. More precisely, the EPL has to be taken as a license with weak copyleft (→ OSLiC, p. 30)
[359] cf. id., l.c., wp. §3 bottom area.
[360] For details of the general OSUC finder → OSLiC, pp. 66 and 69




### 6.6.1 EPL-C1: Using the software only for yourself

**means** that you are going to use a received EPL licensed software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[361]

**requires** no tasks in order to fulfill the conditions of the EPL 1.0 license with respect to this use case:

- You are allowed to use any kind of EPL software in any sense and in any context without being obliged to do anything as long as you do not give the software to 3rd parties.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

[361] For details → OSLiC, pp. 73 - 80

### 6.6.2 EPL-C2: Passing the unmodified software as source code

**means** that you are going to distribute an unmodified version of the received EPL software to 3rd parties – in the form of source code files or as a source code package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02S, OSUC-05S, OSUC-07S[362]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Give the recipient a copy of the EPL 1.0 license. If it is not already part of the software package, add it[363]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the EPL, additionally insert your own correct EPL licensing file.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EPL 1.0 license.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

### 6.6.3 EPL-C3: Passing the unmodified software as binaries

**means** that you are going to distribute an unmodified version of the received EPL software to 3rd parties – in the form of binary files or as a binary package. In this case it is not discriminating to distribute a program, an application,

[362] For details → OSLiC, pp. 73 - 78
[363] For implementing the handover of files correctly → OSLiC, p. 83

a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02B, OSUC-05B, OSUC-07B[364]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file.
- [**mandatory:**] Make the source code of the software accessible via a repository under your own control – even if you did not modified it: Push the source code package into an internet repository and enable its download function. Integrate an easily to find description into your distribution package which explains how the code can be received from where. Ensure, that this repository is usable reasonably long enough.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EPL-C2[365].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EPL 1.0 license.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

[364] For details → OSLiC, pp. 74 - 78

[365] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

### 6.6.4 EPL-C4: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received EPL licensed program, application, or server (proapse) to 3rd parties – in the form of source code files or as a source code package.

**covers** OSUC-04S[366]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**mandatory:**] Mark all modifications of the source code of the program (proapse) thoroughly – namely within the modfied source code.
- [**mandatory:**] Give the recipient a copy of the EPL 1.0 license. If it is not already part of the software package, add it[367]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the EPL, additionally insert your own correct EPL licensing file.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing EPL licensing statements. If you add new source code files, insert a header containing your copyright line and an EPL adequate licensing the statement.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file. Update an existing copyright screen presented by the program so that it shows the same information.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EPL 1.0 license.

[366] For details → OSLiC, pp. 75

[367] For implementing the handover of files correctly → OSLiC, p. 83

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

### 6.6.5 EPL-C5: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received EPL licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[368]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**mandatory:**] Mark all modifications of the source code of the program (proapse) thoroughly – namely within the modfied source code.
- [**mandatory:**] Organize your modifications in a way that they are covered by the (existing) EPL licensing statements.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file. Update an existing copyright screen presented by the program so that it shows the same information.
- [**mandatory:**] Make the source code of the modifed program accessible via a repository under your own control: Push the source code package into an internet repository and enable its download function. Integrate an easily to find description into your distribution package

[368] For details → OSLiC, pp. 75

which explains how the code can be received from where. Ensure, that this repository is usable reasonably long enough.

- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EPL-C4[369].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EPL 1.0 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

### 6.6.6 EPL-C6: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received EPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[370]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**mandatory:**] Mark all modifications of the source code of the library (snimoli) thoroughly – namely within the modfied source code.
- [**mandatory:**] Give the recipient a copy of the EPL 1.0 license. If it is not already part of the software package, add it[371]. If the licensing

[369] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[370] For details → OSLiC, pp. 79

[371] For implementing the handover of files correctly → OSLiC, p. 83

statement in the licensing file of the package does still not clearly state that the package is licensed under the EPL, additionally insert your own correct EPL licensing file.

- [**mandatory:**] Organize your modifications in a way that they are covered by the existing EPL licensing statements. If you add new source code files, insert a header containing your copyright line and an EPL adequate licensing the statement.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EPL 1.0 license.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

### 6.6.7 EPL-C7: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received EPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[372]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution.

[372] For details → OSLiC, pp. 79

- [**mandatory:**] Mark all modifications of the source code of the library (snimoli) thoroughly – namely within the modfied source code.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing EPL licensing statements.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file.
- [**mandatory:**] Make the source code of the modified library accessible via a repository under your own control: Push the source code package into an internet repository and enable its download function. Integrate an easily to find description into your distribution package which explains how the code can be received from where. Ensure, that this repository is usable reasonably long enough.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EPL-6[373].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EPL 1.0 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

### 6.6.8 EPL-C8: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received EPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[374]

[373] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[374] For details → OSLiC, pp. 81

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**mandatory:**] Mark all modifications of the source code of the embedded library (snimoli) thoroughly – namely within the source code.
- [**mandatory:**] Give the recipient a copy of the EPL 1.0 license. If it is not already part of the software package, add it[375]. If the licensing statement in the licensing file of the package does still not clearly state that the embedded library is licensed under the EPL, additionally insert your own correct EPL licensing file.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file. Let the copyright screen of your own overarching program show the same information – as a specification for the embedded component.
- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing EPL licensing statements. If you add new source code files into the scope of the library, insert a header containing your copyright line and an EPL adequate licensing the statement.
- [**voluntary:**] Arrange your source code distribution so that the integrated EPL and the *licensing files* clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the embedded components like libraries, modules, snippets, or plugins in specific directory which contains also all additional licensing elements.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the name of the used EPL licensed component, a link to its homepage, and a link to the EPL 1.0 license

[375] For implementing the handover of files correctly → OSLiC, p. 83

– especially as subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

### 6.6.9 EPL-C9: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received EPL licensed code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[376]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices and the disclaimer of warranty and liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**mandatory:**] Mark all modifications of the source code of the embedded library (snimoli) thoroughly – namely within the source code.
- [**mandatory:**] If still not existing, integrate an explicit, very prominently placed 'No warranty' statement into the distributed source code package. Let this statement clearly say that all (other) contributors to the software do not take over any responsibility for the quality of the software. Then, add the no-warranty clause and the disclaimer of the liability of the EPL itself into that file. Let the copyright screen of your own overarching program show the same information – as a specification for the embedded component.
- [**mandatory:**] Make the source code of the embedded library accessible via a repository under your own control: Push the source code package into an internet repository and enable its download function. Integrate an easily to find description into your distribution package

[376] For details → OSLiC, pp. 81

which explains how the code can be received from where. Ensure, that this repository is usable reasonably long enough.

- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EPL-C8[377].
- [**voluntary:**] Arrange your binary distribution so that the integrated EPL and the *licensing files* clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the embedded components like libraries, modules, snippets, or plugins in specific directory which contains also all additional licensing elements.
- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing EPL licensing statements.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the name of the used EPL licensed component, a link to its homepage, and a link to the EPL 1.0 license – especially as subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any copyright notices contained within the received software package.

### 6.6.10 Discussions and Explanations

The EPL offers a lean section "Requirements"[378] completed by some definitions concerning a "Commercial Distribution"[379]: First it describes, what a distributor must do for correctly distributing an Eclipse licensed program as a set of binaries. Then it describes, what must be done for compliantly distributing the software as source code. Finally it lists two conditions which must be fulfilled in any case[380]. With respect to this structure, we can detect the following tasks:

- The EPL generally requires that "Contributors may not remove or alter any copyright notices contained within the Program"[381] whereas – on the one hand – the word 'Contributor' has to be read as "any person or entity that

[377] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.
[378] cf. *Open Source Initiative*: EPL-1.0, 2005, wp. §3.
[379] cf. id., l.c., wp. §4.
[380] cf. id., l.c., wp. §3.
[381] cf. id., ibid.

distributes the Program", and – on the other hand – the word 'Program' denotes the "initial contribution" and all its modifications[382]. Similar to the EUPL and at least in a very strict reading, also the EPL does not limit these requirements to the distribution of the software (*2others*). But practically, it will be difficult to control the compliant use of the software in those cases where one uses the software only for oneself. But in opposite to – for example – the EUPL, the EPL clearly contains this interdiction. The OSLiC solves this practical inconsistence duplicating the message: On the one hand, it rewrites the negative condition as a mandatory positive assertion for the *2others* use cases (EPL-C2 - EPL-C9). This should emphasize the *activity* to retain the copyright notes in exact the form one has received them. On the other hand, the OSLiC inserts the interdictios into the 'prohibits' section of the *4yourself* use cases (EPL-C1 - EPL-C9).

- Furthermore, the EPL requires that "each Contributor must identify itself as the originator of its Contributions [...] in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution"[383], In this case, 'Contribution' has to be read as the "initial code and documention" together with all subsequent modifications of these parts[384]. For fulfilling this condition faithfully, a developer must mark and describe his modifications of a source code within this source code; and the distributor must describe these modifications on the more general level of software features in a file sometimes called CHANGES. On a first glance, the requirement to document the source code modifications within the source code seems to be restricted to the use cases which concern the distribution of a modified EPL software in the form of source code. But the EPL allows the distribution in the form of binaries only if the distributor also states where one can obtain the correspoding code[385]. So, distributing the binaries implies the distribution of the source code. Therefore the OSLiC inserts the two requirements as mandatory clauses into all the use cases concerning the distribution of a modified EPL software (EPL-C4 - EPL-C9).
- For all distributions in the form of source code the EPL requires that the software "[...] must be made available under this (Eclipse Public License 1.0) Agreement" and that "[...] a copy of this Agreement must be included with each copy of the Program"[386]. Thus, the OSLiC inserts a respective mandatory clause into the use cases (EPL-C4, EPL-C6, EPL-C8). But the EPL is a license with a weak copyleft[387]. Therefore, this conditions does not

[382] cf. *Open Source Initiative*: EPL-1.0, 2005, wp. §1.
[383] cf. id., l.c., wp. §3.
[384] cf. id., l.c., wp. §1.
[385] cf. id., l.c., wp. §3.
[386] cf. id., ibid.
[387] (→ OSLiC, p. 30)

cover the overarching program which uses the embedded library (EPL-C8)

- Additionally, the EPL allows to distribute the software in the form of binaries if the distributor "[...] effectively disclaims on behalf of all Contributors all warranties and conditions [...] (and) effectively excludes on behalf of all Contributorsall liability for damages [...]" – namely in a very broad sense[388]. This delimitation is very important for the EPL. Thus, it subspecifies and explains this aspect once more in a special section titled "Commercial Distribution". There, this aspect is no longer only focussed on a distribution in the form of binaries[389]. So the OSLiC inserts a mandatory clause into all use cases concerning the distribution that the paragraph of "No Warranty"[390] and the "Disclaimer of Liability"[391] of the EPL must explicitly be presented in and by the documentation of distribution package and – if technically possible – by the copyright screen.
- Aside from that, the EPL allows the distribution of the software in the form of binaries only if the distributor clearly "[...] states that the source code for the program is available from such Contributor (distributor) [...]" and if he additionally "[...] informs licensees how to obtain it in a reasonable manner [...]"[392]. This requirement can only be fulfilled seriously if the distributor himself offers the source code via repository. It is not sufficient to point to any external download repository in the world wide web. Thus, – for all use cases concerning the distribution in the form of binaries – the OSLiC follows the respective requirement introduced by the EPL (EPL-C3, EPL-C5, EPL-C7, EPL-C9).
- Finally, one has clearly to state that this rule above evokes a real source code distribution which therefore must follow the rules of distributing the software. Thus, the OSLiC requires in all cases of a binary distribution to execute also the task-lists of the respective source code use cases.

## 6.7 EUPL licensed software

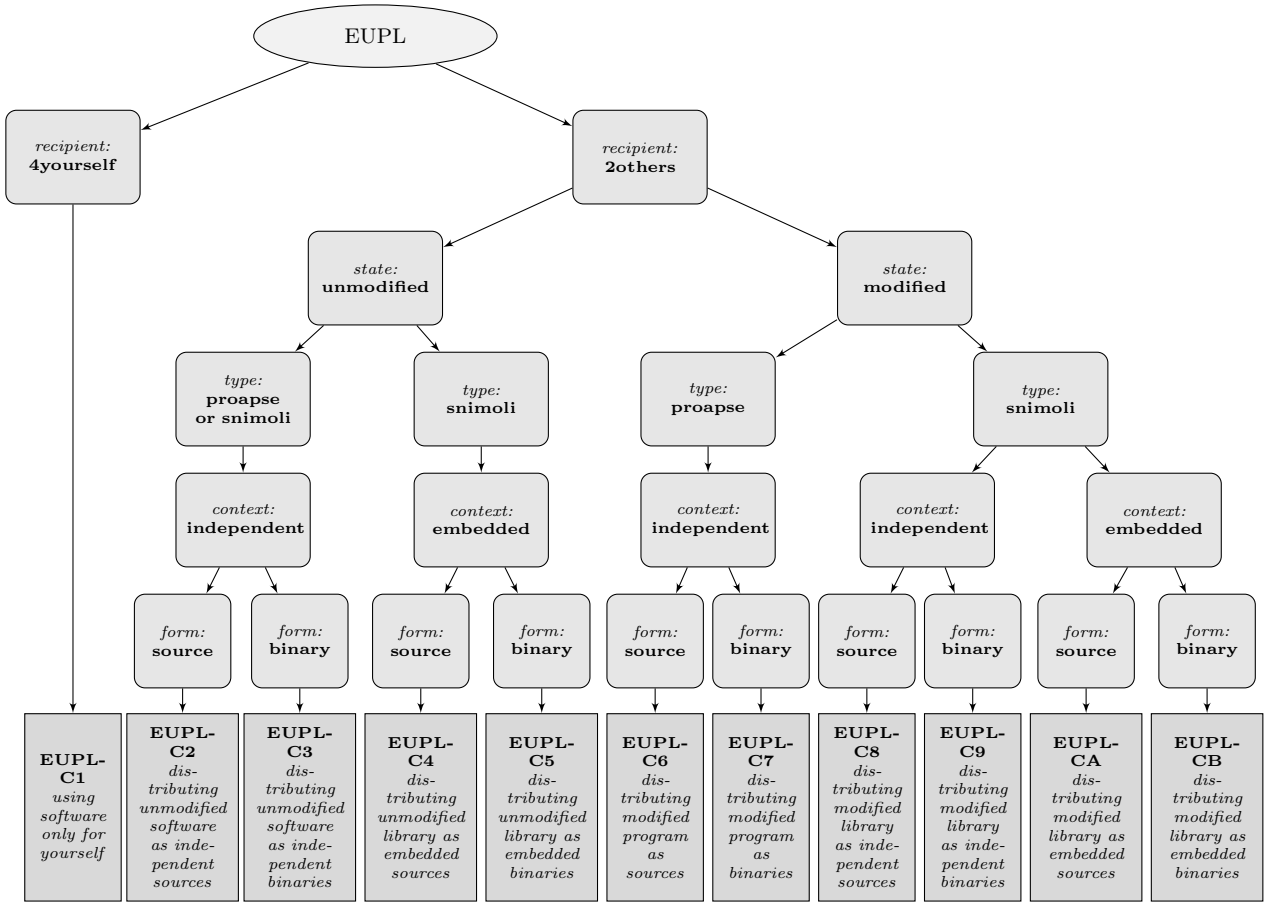
The European Union Public License explicitly distinguishes the distribution of the source code from that of the binaries: In the chapter "Communication of the Source Code", it allows to "provide the Work either in its Source Code form, or as Executable Code"[393]. But if a piece of EUPL licensed software is distributed

[388] cf. *Open Source Initiative*: EPL-1.0, 2005, wp. §3.
[389] cf. id., l.c., wp. §4.
[390] cf. id., l.c., wp. §5.
[391] cf. id., l.c., wp. §6.
[392] cf. id., l.c., wp. §3.
[393] cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp. §3.

as binary package, then the license additionally requires that the distributor either "[...] provides a machine-readable copy of the Source Code [...]" directly together with the binaries[394] or that he "[...] indicates [...] a repository where the Source Code is easily and freely accessible for as long as the Licensor continues to distribute [...] the Work"[395]. For respecting this conditions it is irrelevant whether the software has been modified or not; and all the other "obligations of the licensee" refer to both forms[396].

But there is a specific aspect which has to be considered for acting in accordance to the EUPL: In the proper meaning of the words, the EUPL is a license with a weak copy left, no doubt. But this character is only evoked by the fact that the EUPL allows the licensee to relicense the software by following the conditions of a specific clause and an licenses listed in an appendix which also lists some licenses with a weak copy left.[397]. Thus, with respect to question how to fulfill the license best, it is safer to treat the EUPL as a license with a default strong copy left. Concerning the use of an unmodified or a modified library as an embedded component, a license with a strong copy left evokes that also the application which is using the (un)modified library has to be licensed under the same conditions as the library itself. Thus, for finding the simply processable task lists, the following EUPL specific and a little more sophisticated open source use case structure[398] can be used:

[394] cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp. §5.
[395] cf. id., l.c., wp. §3.
[396] cf. id., l.c., wp. §5.
[397] (→ OSLiC, p. 31)
[398] For details of the general OSUC finder → OSLiC, pp. 66 and 69




### 6.7.1 EUPL-C1: Using the software only for yourself

**means** that you are going to use a received EUPL licensed software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[399]

**requires** no tasks in order to fulfill the conditions of the EUPL 1.1 license with respect to this use case:

- You are allowed to use any kind of EUPL software in any sense and in any context without being obliged to do anything as long as you do not give the software to 3rd parties.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this

[399] For details → OSLiC, pp. 73 - 80

EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.2 EUPL-C2: Passing the unmodified software as independent sources

**means** that you are going to distribute an unmodified version of the received EUPL software to 3rd parties – as an independent unit and in the form of source code files or as a source code package. In this case, it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin.

**covers** OSUC-02S, OSUC-05S[400]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the software package, add it[401].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.3 EUPL-C3: Passing the unmodified software as independent binaries

**means** that you are going to distribute an unmodified version of the received EUPL software to 3rd parties – as an independent unit and in the form of binary files or as a binary package. In this case, it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin.

**covers** OSUC-02B, OSUC-05B[402]

[400] For details → OSLiC, pp. 73 - 76
[401] For implementing the handover of files correctly → OSLiC, p. 83
[402] For details → OSLiC, pp. 74 - 77

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the binary package, add it[403].
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control (even if you do not modified it): Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for as long as you continue to distribute the software.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EUPL-C2[404].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.4 EUPL-C4: Passing the unmodified library as embedded sources

**means** that you are going to distribute an unmodified version of the received EUPL licensed snippet, module or library to 3rd parties – as embedded component of a larger unit and in the form of source code files or as a source code package.

[403] For implementing the handover of files correctly → OSLiC, p. 83

[404] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

**covers** OSUC-07S[405]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the software package, add it[406].
- [**mandatory:**] License your overarching program also under the EUPL 1.1; Organize the sources of the on-top development in a way that they are also covered by the EUPL-1.1 licensing statements.
- [**voluntary:**] Let the copyright dialog of the on-top development clearly say, that it uses the EUPL-1.1 licensed library and that it is itself licensed under the EUPL-1.1 too.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.5 EUPL-C5: Passing the unmodified library as embedded binaries

**means** that you are going to distribute an unmodified version of the received EUPL licensed snippet, module or library to 3rd parties – as embedded component of a larger unit and in the form of binary files or as a binary package.

**covers** OSUC-07B[407]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license

[405] For details → OSLiC, pp. **??**
[406] For implementing the handover of files correctly → OSLiC, p. 83
[407] For details → OSLiC, pp. 78

and to the disclaimer of warranties – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.

- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the binary package, add it[408].
- [**mandatory:**] Make the source code of the embedded library ***and*** the source code of your overarching program accessible via a repository under your own control (even if you do not modified it): Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for as long as you continue to distribute the software.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] License your overarching program also under the EUPL 1.1: Organize the binaries of the on-top development in a way that they are also covered by the EUPL-1.1 licensing statements.
- [**mandatory:**] Execute the to-do list of use case EUPL-C4[409].
- [**voluntary:**] Let the copyright dialog of the on-top development clearly say, that it uses the EUPL-1.1 licensed library and that it is itself licensed under the EUPL-1.1 too.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

[408] For implementing the handover of files correctly → OSLiC, p. 83

[409] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

### 6.7.6 EUPL-C6: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received EUPL licensed program, application, or server (proapse) to 3rd parties – in the form of source code files or as a source code package.

**covers** OSUC-04S[410]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the software package, add it[411].
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**mandatory:**] Mark all modifications of source code of the program (proapse) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing EUPL licensing statements. If you add new source code files, insert a header containing your copyright line and an EUPL adequate licensing the statement.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

[410] For details → OSLiC, pp. 75
[411] For implementing the handover of files correctly → OSLiC, p. 83

### 6.7.7 EUPL-C7: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received EUPL licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[412]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the binary package, add it[413].
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing EUPL licensing statements.
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control: Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for as long as you continue to distribute the software.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EUPL-C6[414].
- [**voluntary:**] Mark all modifications of source code of the program (proapse) thoroughly – namely within the source code and including the date of the modification.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright*

[412] For details → OSLiC, pp. 75

[413] For implementing the handover of files correctly → OSLiC, p. 83

[414] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

*notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.8 EUPL-C8: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received EUPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[415]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the software package, add it[416].
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**mandatory:**] Mark all modifications of source code of the library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing EUPL licensing statements. If you add new source code files, insert a header containing your copyright line and an EUPL adequate licensing the statement.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright*

[415] For details → OSLiC, pp. 79
[416] For implementing the handover of files correctly → OSLiC, p. 83

*notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.9 EUPL-C9: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received EUPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[417]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them. If you compile the binary from the sources, ensure that all the licensing elements are also incorporated into the package.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the binary package, add it[418].
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing EUPL licensing statements.
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control: Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for as long as you continue to distribute the software.

[417] For details → OSLiC, pp. 79

[418] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EUPL-C8[419].
- [**voluntary:**] Mark all modifications of source code of the library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the EUPL 1.1 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.10 EUPL-CA: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received EUPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[420]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them.
- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the software package, add it[421].
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.

[419] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[420] For details → OSLiC, pp. 81

[421] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing EUPL licensing statements. If you add new source code files into the scope of the library, insert a header containing your copyright line and an EUPL adequate licensing the statement.
- [**mandatory:**] License your overarching program also under the EUPL 1.1.
- [**mandatory:**] Mark all modifications of source code of the embedded library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**voluntary:**] Let the copyright dialog of the on-top development clearly say, that it uses the EUPL-1.1 licensed library and that it is itself licensed under the EUPL-1.1 too.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the name of the used EUPL licensed component, a link to its homepage, and a link to the EUPL 1.1 license – especially as subsection of your own copyright notice.

**prohibits** ...

- to promote any of your services or products – based on the this software – by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.11 EUPL-CB: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received EUPL licensed code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[422]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the copyright, patent or trademarks notices and all notices that refer to the license and to the disclaimer of warranties – are retained in your package in the form you have received them. If you compile the binary from the

[422] For details → OSLiC, pp. 81

sources, ensure that all the licensing elements are also incorporated into the package.

- [**mandatory:**] Give the recipient a copy of the EUPL 1.1 license. If it is not already part of the binary package, add it[423].
- [**mandatory:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**mandatory:**] Make the source code of the embedded library and the source code of your overarching program accessible via a repository under your own control [424]: Push the source code package into a repository and make it downloadable via the internet. Integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for as long as you continue to distribute the software.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case EUPL-CA[425].
- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing EUPL licensing statements.
- [**mandatory:**] License your overarching program also under the EUPL 1.1.
- [**voluntary:**] Mark all modifications of source code of the embedded library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the name of the used EUPL licensed component, a link to its homepage, and a link to the EUPL 1.1 license – especially as subsection of your own copyright notice.

**prohibits** ...

- to promote any of your services or products – based on the this software

[423] For implementing the handover of files correctly → OSLiC, p. 83

[424] Formally, the EUPL-1.1 is only a license with weak copyleft. But this is only a result of the allowance to relicense the software (→ OSLiC, p. 31). So, as long as you do not relicense the embedded library with respect to the list of "compatible licenses accodring to article 5 EUPL" (cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp §5 and Appendix), you also have to publish the code of your overarching work.

[425] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

– by trade names, trademarks, service marks, or names linked to this EUPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.7.12 Discussions and Explanations

- The EUPL generally "[...] does not grant permission to use the trade names, trademarks, service marks, or names of the Licensor , except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the copyright notice"[426]. Therefore, the OSLiC genreally interdicts (EUPL-C1 - EUPL-CB) to promote any service or product – based on this software – by such elements.
- The EUPL generally requires that "[...] the Licensee shall keep intact all copyright, patent or trademarks notices and all notices that refer to the Licence and to the disclaimer of warranties"[427]. In a very strict reading, the EUPL does not limit this requirement to the distribution of the software. But practically, it will be impossible to control the compliant use of the software in those cases (*4yourself*) without 'switching' into the use case 'distribution'. Therefore the OSLiC only inserts this requirement as a mandatory clause for the *2others* use cases (EUPL-C2 - EUPL-CB).
- The EUPL also requires to "[...] include [...] a copy of the (EUPL) Licence with every (distributed) copy of the Work [...]"[428]. Therefore, all *2others* use cases contain the respective mandatory condition (EUPL-C2 - EUPL-CB).
- Additionally, the EUPL requires, that the "licensee" who distribiutes a modified work "[...] must cause any Derivative Work to carry prominent notices stating that the Work has been modified and the date of modification"[429]. Thus, the OSLiC integrates the mandatory requirement to generate (update) a respective notice file into all 'modification use cases and recommends to mark all modifications in the source code (EUPL-C6 - EUPL-CB)
- Furthermore, the EUPL requires that any distributor of the software "[...] provide a machine-readable copy of the Source Code [...]" by "[...] (indicating) a repository where this Source will be easily and freely available for as long as the Licensee continues to distribute [...] the Work"[430]. Therefore the OSLiC inserts a respective requirement into the task list of all

[426] cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp. §5.
[427] cf. id., ibid.
[428] cf. id., ibid.
[429] cf. id., ibid.
[430] cf. id., ibid.. To be precise, the EUPL also allows to directly distribute the source code to-

cases concerning a binary distribution (EUPL-C3, EUPL-C7, EUPL-C9, EUPL-CB)

- Finally, the EUPL contains a "copyleft clause" stating that if a "[...] Licensee distributes [...] copies of the Original Works or Derivative Works based upon the Original Work, this Distribution [...] will be done under the terms of this (EUPL) Licence [...]". In all the use cases which do not concern the use of an embedded component (EUPL-C2 - EUPL-C9) this copyleft clause is already fulfilled by either distributing the modified sources themselves of by making them accessible via a repository. In those cases where the licensee distributes an overarching program which uses an EUPL licensed component (EUPL-CA - EUPL-CB) normally also the code of the overarching work must be distributed. Thus, with respect to the use case (EUPL-CA) this is already fulfilled by definition. So, the OSLiC only mentions this default view in the case EUPL-CB and therefore implicitly evokes a strong copyleft effect of the EUPL[431].

## 6.8 GPL licensed software

Both versions of the GNU General Public License explicitly distinguish the distribution of the source code from that of the binaries: On the one hand, the GPL-V2 mainly talks about copying and distributing the source code[432], but also mentions the specific conditions for "[...] (copying) and (distributing) the Program [...] in object code or executable form [...]"[433]. On the other hand, the GPL-V3 describes the "Basic Permissions" and the conditions for "Conveying Verbatim Copies" or for "Conveying Modified Source Versions"[434] before it explains the rules for "Conveying Non-Source-Forms"[435]. Additionally, GPL-V2 and GPL-V3 mainly talk about copying *and* distributing the software; the private use is nearly complete unspecified[436]. Finally, the GPL-V2 and the GPL-V3 are aiming for the

gether with the binary packages (cf. *European Community a. European commission Joinup*: EUPL-1.1/EN, 2007, wp. §3). With respect to the OSLiC principle to offer only one reliable way, the OSLiC simplifies this option: It 'only' asks for the repository solution.

431) Formally, the EUPL-1.1 is a license with weak copyleft. But this is only a result of the allowance to relicense the software (→ OSLiC, p. 31). So, as long as you do not relicense the embedded library with respect to the list of "compatible licenses according to article 5 EUPL", you also have to publish the code of your overarching work. Therefore, you can only avoid this consequence by relicensing the embedded component by one of the compatible licenses with a weak copyleft listed in the EUPL appendix (cf. id., l.c., wp §5 and Appendix)

432) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §1, §2.

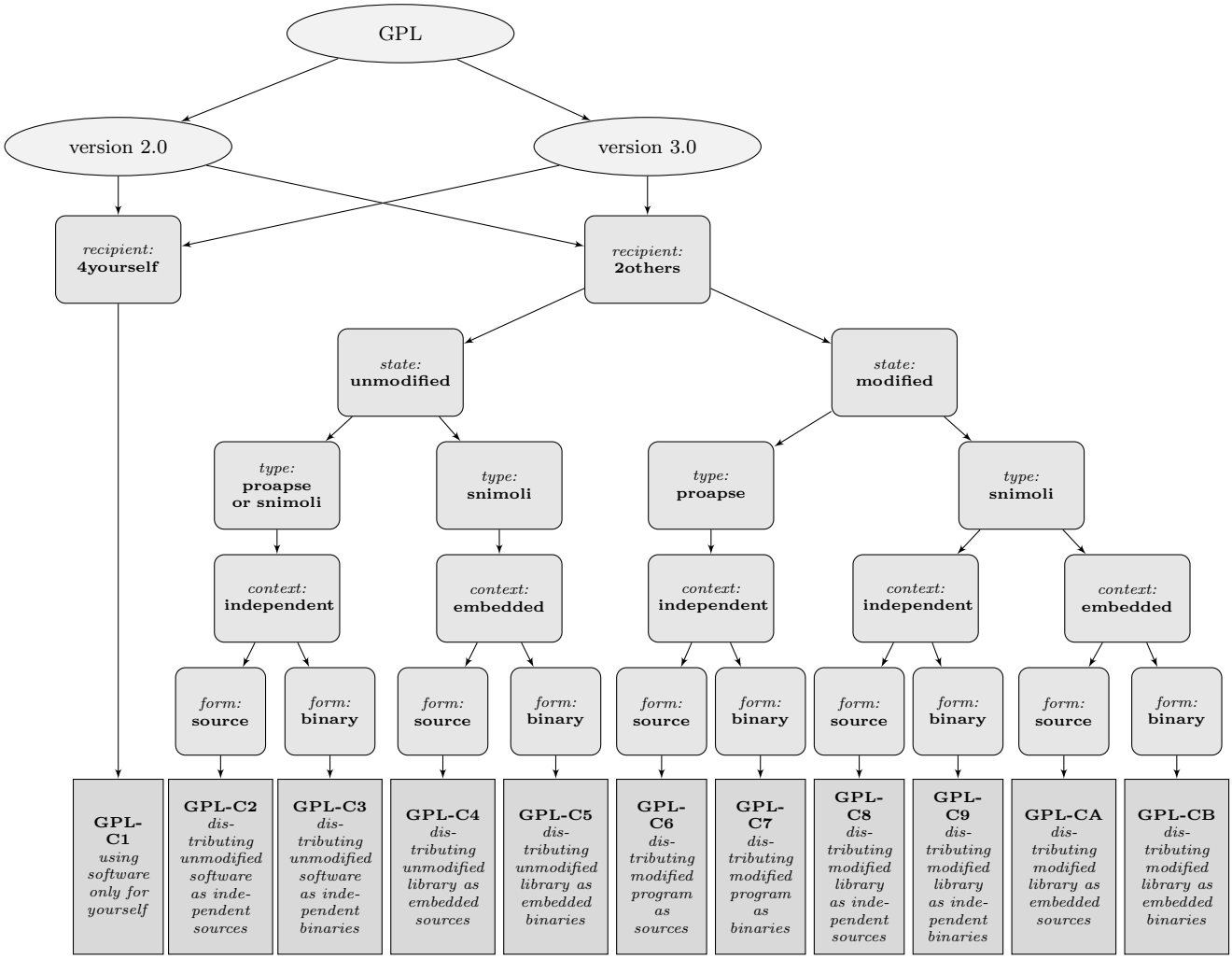
433) cf. id., l.c., wp. §3.

434) cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §2, §4, §5.

435) cf. id., ibid.

436) The GPL-V2 lists its 'restrictions' only with respect to the act of copying *and* distributing "copies of the program" (cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp.

same results and the same spirit by requiring nearly the same license fulfilling tasks. Last but not least, with respect to the use of an unmodified or a modified library as an embedded component, a license with a strong copy left evokes that also the application which is using the (un)modified library has to be licensed under the same conditions as the library itself. Therefore it is appropriate to cover both versions by the same chapter and to offer the same more sophisticated GPL specific open source use case structure[437] for finding the easily processable corresponding task lists:




§1, §2, §4 et passim; emphasizings by KR) while the GPL-V3 explicitly specifies that one "[...] may make, run and propagate covered works that (one does) not convey, without conditions so long as (the) license otherwise remains in force" (cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §2).

437) For details of the general OSUC finder → OSLiC, pp. 66 and 69

### 6.8.1 GPL-C1: Using the software only for yourself

**means** that you are going to use a received GPL-V2 or GPL-V3 licensed software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[438]

**requires** no tasks in order to fulfill the conditions of the GPL-V2 or the GPL-V3 with respect to this use case:

- You are allowed to use any kind of GPL software in any sense and in any context without being obliged to do anything as long as you do not give the software to 3rd parties.

**prohibits** nothing explictly with respect to this use case.

### 6.8.2 GPL-C2: Passing the unmodified software as independent sources

**means** that you are going to distribute an unmodified version of the received GPL-V2 or GPL-V3 licensed software to 3rd parties – as an independent unit and in the form of source code files or as a source code package. In this case, it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin.

**covers** OSUC-02S, OSUC-05S[439]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed source code package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[440].

[438] For details → OSLiC, pp. 73 - 80
[439] For details → OSLiC, pp. 73 - 76
[440] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Retain all existing copyright notices.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing explictly with respect to this use case.

### 6.8.3 GPL-C3: Passing the unmodified software as independent binaries

**means** that you are going to distribute an unmodified version of the received GPL-V2 or GPL-V3 licensed software to 3rd parties – as an independent unit and in the form of binary files or as a binary package. In this case, it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin.

**covers** OSUC-02B, OSUC-05B[441]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[442].
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control (even if you do not modified it): Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can

[441] For details → OSLiC, pp. 74 - 77
[442] For implementing the handover of files correctly → OSLiC, p. 83

be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.

- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Execute the to-do list of use case GPL-C2[443].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing explictly with respect to this use case.

### 6.8.4 GPL-C4: Passing the unmodified library as embedded sources

**means** that you are going to distribute an unmodified version of the received GPL-V2 or GPL-V3 licensed snippet, module or library to 3rd parties – as embedded component of a larger unit and in the form of source code files or as a source code package.

**covers** OSUC-07S[444]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed source code package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)

[443] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[444] For details → OSLiC, pp. 78

- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[445].
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Let the copyright dialog of the on-top development clearly say, that it uses the GPL licensed library and that it is itself licensed under the GPL-V2 resp. GPL-V3 too. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.
- [**mandatory:**] Organize the sources of the on-top development in a way that they are also covered by the GPL-V2 resp. GPL-V3 licensing statements.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing explictly with respect to this use case.

### 6.8.5 GPL-C5: Passing the unmodified library as embedded binaries

**means** that you are going to distribute an unmodified version of the received GPL-V2 or GPL-V3 licensed snippet, module or library to 3rd parties – as embedded component of a larger unit and in the form of binary files or as a binary package.

**covers** OSUC-07B[446]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the

[445] For implementing the handover of files correctly → OSLiC, p. 83
[446] For details → OSLiC, pp. 78

license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)

- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[447].
- [**mandatory:**] Make the source code of the embedded library ***and*** the source code of your overarching program accessible via a repository under your own control: Push the source code package into a repository and make it downloadable via the internet. Integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Let the copyright dialog of the on-top development clearly say, that it uses the GPL licensed library and that it is itself licensed under the GPL-V2 resp. GPL-V3 too. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.
- [**mandatory:**] Organize the binaries of the on-top development in a way that they are also covered by the GPL-V2 resp. GPL-V3 licensing statements.
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Execute the to-do list of use case GPL-C4[448].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the used software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing explictly with respect to this use case.

[447] For implementing the handover of files correctly → OSLiC, p. 83

[448] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

### 6.8.6 GPL-C6: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received GPL-V2 or GPL-V3 licensed program, application, or server (proapse) to 3rd parties – in the form of source code files or as a source code package.

**covers** OSUC-04S[449]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 resp. GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed source code package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 or GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license.
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Let the copyright dialog of the program clearly say that it is a GPL licensed program. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3. If these conditions are not already fulfilled, add the missed elements.
- [**mandatory:**] Mark all modifications of source code of the program (proapse) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing GPL licensing statements. If you add new source code files, insert a header containing your copyright line and a licensing the statement in the form required by the GNU project[450].

[449] For details → OSLiC, pp. 75

[450] For details see section 'How to Apply These Terms to Your New Programs' in the GPL-v2 resp. GPL-v3 license.

- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications on a more functional level.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V3 resp. to the GPL-V3.

**prohibits** nothing explictly with respect to this use case.

### 6.8.7 GPL-C7: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received GPL-V2 or GPL-V3 licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[451]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2resp. the GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 resp. GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[452].
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Mark all modifications of source code of the program (proapse) thoroughly namely within the source code and including the date of the modification.

[451] For details → OSLiC, pp. 75
[452] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Let the copyright dialog of the program clearly say that it is a GPL licensed program. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3. If these conditions are not already fulfilled, add the missed elements.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing GPL licensing statements.
- [**mandatory:**] Make the source code of the distributed software accessible via a via a repository under your own control: Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case GPL-C6[453].
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications on a more functional level.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. to the GPL-V3.

**prohibits** nothing explictly with respect to this use case.

### 6.8.8 GPL-C8: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received GPL-V2 or GPL-V3 licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[454]

[453] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[454] For details → OSLiC, pp. 79

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed source code package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[455].
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Mark all modifications of source code of the library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing GPL licensing statements. If you add new source code files, insert a header containing your copyright line and a licensing the statement in the form required by the GNU project[456].
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing with respect to this use case.

[455] For implementing the handover of files correctly → OSLiC, p. 83

[456] For details see section 'How to Apply These Terms to Your New Programs' in the GPL-v2 resp. GPL-v3 license.

### 6.8.9 GPL-C9: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received GPL-V2 or GPL-V3 licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[457]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[458].
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control: Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case GPL-C8[459].
- [**mandatory:**] Mark all modifications of source code of the library (snimoli) thoroughly – namely within the source code and including

[457] For details → OSLiC, pp. 79

[458] For implementing the handover of files correctly → OSLiC, p. 83

[459] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

the date of the modification.

- [**mandatory:**] Organize your modifications in a way that they are covered by the existing GPL licensing statements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing with respect to this use case.

### 6.8.10 GPL-CA: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received GPL-V2 or GPL-V3 licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[460]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed source code package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[461].

[460] For details → OSLiC, pp. 81

[461] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Let the copyright dialog of the on-top development clearly say, that it uses the GPL licensed library and that it is itself licensed under the GPL-V2 resp. GPL-V3 too. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.
- [**mandatory:**] Mark all modifications of source code of the embedded library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing GPL licensing statements. If you add new source code files to library itself, insert a header containing your copyright line and a licensing the statement in the form required by the GNU project[462].
- [**mandatory:**] Organize the sources of the on-top development in a way that they are also covered by the GPL-V2 resp. GPL-V3 licensing statements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing with respect to this use case.

### 6.8.11 GPL-CB: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received GPL-V2 or GPL-V3 licensed code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[463]

[462] For details see section 'How to Apply These Terms to Your New Programs' in the GPL-v2 resp. GPL-v3 license.

[463] For details → OSLiC, pp. 81

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the GPL-V2 or GPL-V3 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license GPL-V2 itself resp. by the GPL-V3 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the GPL-V2 resp. GPL-V3 license. If it is not already part of the software package, add it[464].
- [**mandatory:**] Retain all existing copyright notices.
- [**mandatory:**] Make the source code of the embedded library and the source code of your overarching program accessible via a repository under your own control: Push the source code package into a repository and make it downloadable via the internet. Integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case GPL-CA[465].
- [**mandatory:**] Let the copyright dialog of the on-top development clearly say, that it uses the GPL licensed library and that it is itself licensed under the GPL-V2 resp. GPL-V3 too. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.
- [**mandatory:**] Mark all modifications of source code of the embedded library (snimoli) thoroughly – namely within the source code and including the date of the modification.

[464] For implementing the handover of files correctly → OSLiC, p. 83

[465] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing GPL licensing statements.
- [**mandatory:**] Organize the binaries of the on-top development in a way that they are also covered by the GPL-V2 resp. GPL-V3 licensing statements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2 resp. GPL-V3.

**prohibits** nothing with respect to this use case.

### 6.8.12 Discussions and Explanations

- The GPL-V2 allows to "[...] to copy *and* (to) distribute verbatim copies of the Program's complete source code as you receive it [...] provided that you [a] conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; [b] keep intact all the notices that refer to this License and to the absence of any warranty; and [c] distribute a copy of this License along with the Program"[466]. Additionally, the GPL-V2 allows to "[...] copy and distribute [...] modifications (of the Program or any portion of it) [...] under the terms of Section 1"[467] while it allows to distribute binaries "under the terms of Sections 1 and 2"[468]. But the GPL does not require any tasks if you are using the work only for yourself. Thus, the quoted conditions of "Section 1" are mandatory for all use cases concerning the distribution of an GPL licensed work (GPL-C2 - GPL-CB)[469].

[466] cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §1, emphasizes by KR.
[467] cf. id., l.c., wp. §2.
[468] cf. id., l.c., wp. §4.
[469] The GPL-V3 uses a similar structure to establish the same requirements: In §4 it allows to "[...] convey verbatim copies of the Program's source code as you receive it [...] provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program". Additionally in §5 it also allows to "[...] convey [...] modifications [...] under the terms of section 4 [...]" and in §6 it allows to "[...] convey a covered work in object form under the terms

- The GPL-V2 allows to "[...] copy and (to) distribute the Program (or a work based on it [...]) in object code or executable form [...] provided that you accompany it with the complete corresponding machine-readable source code [...] on a medium customarily used for software interchange"[470]. As a substitution for this basic condition, the GPL-V2 allows to "accompany" the binary distribution package "[...] with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corryponding source code [...] on a medium customarily used for software interchange"[471]. The OSLiC construes the common technique to download files from the internet as a distribution *on a medium [being today] customarily used for software interchange*. Therefore, the OSLiC requires for all open source use cases which refer to the distribution of binaries (GPL-C3, GPL-C7, GPL-C9, GPL-CB) to make the corresonding source code of the library itself accessible via an internet repository under your own control[472]. The weakness that in this case "third parties [which have received the binaries] are not compelled to copy the source code [...]" is mediately accepted by the GPL[473]. But the necessity to offer the source code via a repository being controled by yourself (mostly) may not be circumvented: The GPL-V2 allows to redistribute a link to an external source code repository only in case of "noncommercial distributions"[474].
- Both, the GPL-V2 and the GPL-V3 allow to "[...] modify your copy or copies of the Program or any portion of it [...] and (to) copy and distribute such modifications [...]" only under very similar restrictions

of sections of 4 and 5" (cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §4, §5, §6). In opposite to the GPL-V2, the GPL-V3 explicitly states that one "[...] may make, run and propagate covered works that (one) (does) not convey [distribute], without conditions so long as (the GPL-V3) license otherwise remains in force" (cf. id., l.c., wp. §2). Moreover, giving a package to a third party for getting a modified version back has not to be taken as a case of distribution if the modification has only been executed on behalf and only for the purpose of the purchaser and if the modified version is not distributed to any third party (cf. id., ibid.). If one collects all these GPL-V3 statements together, than one may conclude that the tasks which fulfill the corresponding GPL-V2 requirements together also fit the GPL-V3 conditions.

470) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §3, §3a.

471) cf. id., l.c., wp. §3b.

472) Also the GPL-V3 explicitly requires to make the source code accessible in case of distributing binaries. But in opposite to the GPL-V2, the GPL-V3 explicitly offers the option that giving the "[...] access to copy the Corresponding Source from a network server at no charge" would fulfill the conditions (cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §6 and §6b). So again, the tasks which ensure to act in accordance to the GPL-V2 license in case of distributing binaries, also fulfill the conditions of the GPL-V3

473) cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §3, at the end.

474) cf. id., l.c., wp. §3c.

and conditions[475]:

- First, modified files must be marked as modifications and marked by the date of the modification[476]. These conditions must be respected by all open source use cases concerning the distribution of the modified work [GPL-C6 - GPL -9], because even if one primarily intends to distribute binaries, one has lateron also to deliver the source code. The OSLiC rewrites this requirement by the mandatory condition to mark each modified file and by the voluntary condition to update / generate a general changing file.
- Second, the GPL requires that all copies of the modified software which are using an interactive interface or a method to display messages must "[...] print or display an announcement including an appropriate copyright notice and a notice that there is no warranty [...] and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License"[477]. The OSLiC rewrites this condition in the form that the work shall let its copyright dialog clearly reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the GPL-V2-file resp. the GPL-V3-file which has to be delivered together with the software. These conditions have to be respected if one redistributes the received and then modified programs (GPL-C6, GPL-C7) or if one distributes own programs which are using (modified) libraries as embedded components (GPL-CA, GPL-CB). For those open source use cases which concern the redistribution of received and modified libraries etc., the OSLiC does not mention these requirements because libraries, plugins, or snippets normally do not offer their own copyright dialogs.
- Third, the GPL requires to " [...] cause any work (being distributed or published), that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this (GPL)"[478]. This requirement does not depend of the form in which the software is distributed. The OSLiC adopts this statement in the following way:
  - For all open source use cases which concern the distribution (GPL-C2 ... GPL-CB), the OSLiC rewrites this condition as the manda-

[475] cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §2.

[476] For GPL-V2 see cf. id., ibid.. For GPL-V3 see cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §5

[477] For GPL-V2 see cf. *Open Source Initiative*: The GPL-2.0 License (OSI), 1991, wp. §2c. For GPL-V3 see cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §5d

[478] For GPL-V2 see cite[cf.][wp. §2b]Gpl20OsiLicense1991a. For GPL-V3 see cite[cf.][wp. §5c]Gpl30OsiLicense2007a

tory requirement to retain all existing licensing elements.

* For all use cases which deal with the distribution of a modified version of the software (GPL-C6 ... GPL-CB), the OSliC adds the requirement to organize the modifications in a way that they are covered by the respective GPL-V2 oder GPL-V3 licensing statements.
* For the use case which deal with the distribution of an embedded library (GPL-C4,GPL-C5,GPL-CA,GPL-CB) the OSLiC requires also to license the on-top development under the terms of the respective GPL-V2 or GPL-V3 license.

– Finally, as parts of those task lists which concern the distribution in the form of binaries, the OSLiC reminds the reader also to execute the corresponding source code use cases because distributing the binaries without making the corresponding sources accessible is not allowed by the GPL.

## 6.9 LGPL licensed software

Both versions of the GNU Lesser General Public License explicitly distinguish the distribution of the source code from that of the binaries: On the one hand, the LGPL-2.1 mainly talks about copying and distributing the source code[479]. But it also directly mentions the specific conditions for "[...] (copying) and (distributing) the Library [...] in object code or executable form [...]"[480]. On the other hand, also the LGPL-3.0 and the GPL-3.0 – which have to be considered together because the GPL-3.0 is included into the LGPL-3.0[481] – are treating the distribution of source code and the distribution of the object code as different aspects of the same phenomenon[482]. Additionally, LGPL-2.1 and LGPL-3.0 mainly talk about copying and distribution the software; the private use is nearly complete unspecified[483]. Finally, the LGPL-2.1 and the LGPL-3.0 are aiming for the same results and the same spirit by requiring nearly the same license fulfilling

[479] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §1, §2, §5, §6.

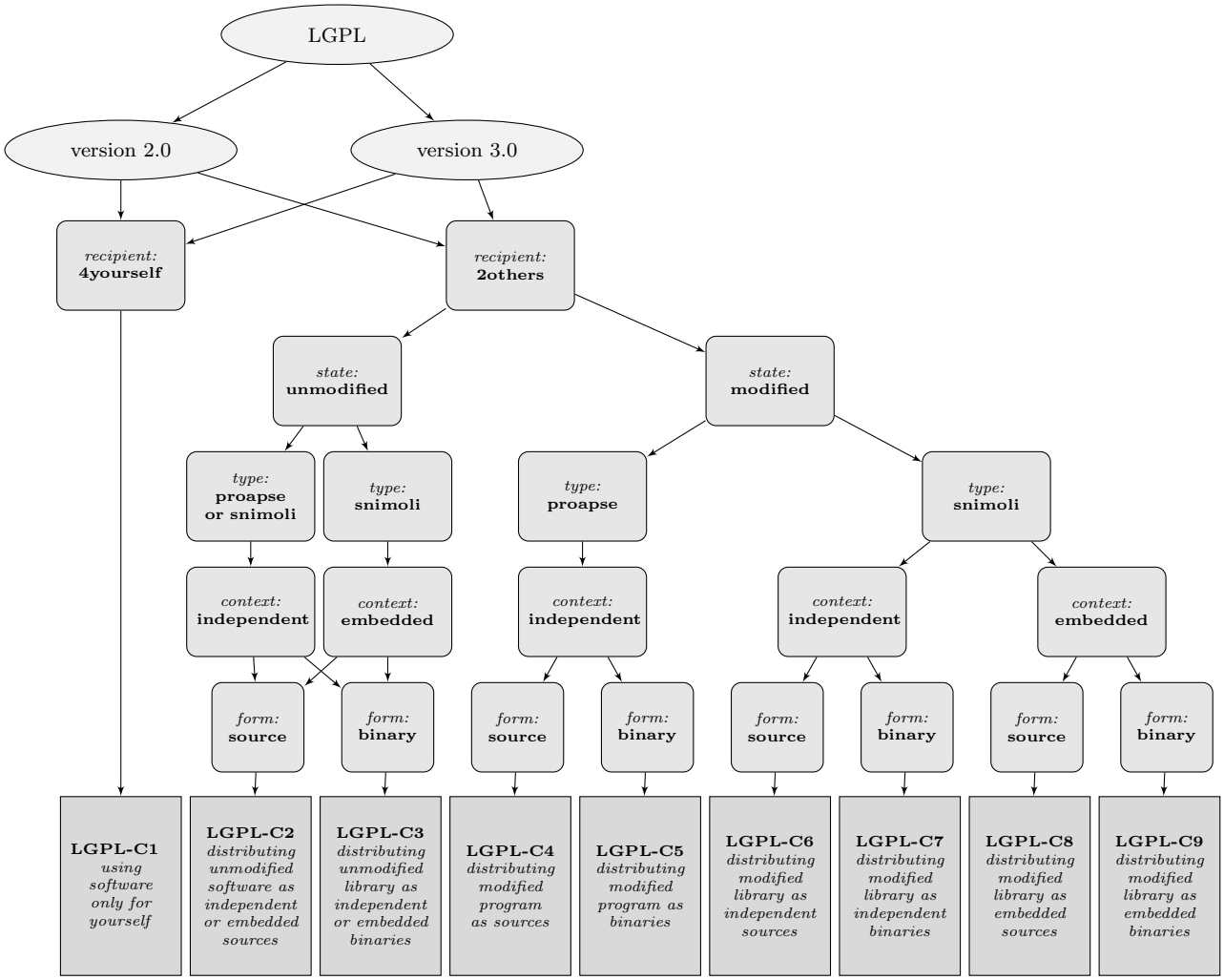
[480] cf. id., l.c., wp. §4.

[481] cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp, just before §0.

[482] The GPL-3.0 contains a specific section named "Conveying Non-Source Forms" which describes the conditions to "[...] convey a covered work in object code form [...]" (cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §6)), while the LGPL-3.0 explicitly deals with the "object code incorporating material from (the) library header files"(cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp. §3)).

[483] The LGPL-2.1 lists its 'restrictions' only with respect to the act of copying and distributing "copies of the library" (cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §1, §2, §4 et passim) while the GPL-3.0 explicitly specifies that one "[...] may make, run and propagate covered works that (one does) not convey, without conditions so long as (the)

tasks. Therefore it is mostly appropriate to cover both versions in one chapter[484] and to offer the same LGPL specific open source use case structure[485] for finding the easily processable corresponding task lists:




### 6.9.1 LGPL-C1: Using the software only for yourself

**means** that you are going to use a received LGPL-v2.1 resp. LGPL-v3 licensed software only for yourself and that you do not hand it over to any 3rd party in any sense.

license otherwise remains in force" (cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §2).

484) The exception concerns the distribution of a modified program, application, or server under the terms of the LGPL

485) For details of the general OSUC finder → OSLiC, pp. 66 and 69

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[486]

**requires** no tasks in order to fulfill the conditions of the LGPL-2.1 or the LGPL-3.0 with respect to this use case:

- You are allowed to use any kind of LGPL-v2.1 resp. LGPL-v3 licensed software in any sense and in any context without being obliged to do anything as long as you do not give the software to 3rd parties.

**prohibits** nothing explictly with respect to this use case.

### 6.9.2 LGPL-C2: Passing the unmodified software as source code

**means** that you are going to distribute an unmodified version of the received LGPL-v2.1 resp. LGPL-v3 licensed software to 3rd parties – in the form of source code files or as a source code package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02S, OSUC-05S, OSUC-07S[487]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-2.1 or LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license LGPL-2.1 itself resp. by the LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the LGPL-2.1 resp. LGPL-3.0 license. If it is not already part of the software package, add it[488].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.

[486] For details → OSLiC, pp. 73 - 80
[487] For details → OSLiC, pp. 73 - 78
[488] For implementing the handover of files correctly → OSLiC, p. 83

- [**voluntary:**] Retain all existing copyright notices.

**prohibits** nothing explictly with respect to this use case.

### 6.9.3 LGPL-C3: Passing the unmodified software as binaries

**means** that you are going to distribute an unmodified version of the received LGPL-v2.1 resp. LGPL-v3 licensed software to 3rd parties – in the form of binary files or as a binary package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02B, OSUC-05B, OSUC-07B[489]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-2.1 or LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license LGPL-2.1 itself resp. by the LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the LGPL-2.1 resp. LGPL-3.0 license. If it is not already part of the software package, add it[490].
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control (even if you do not modified it): Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.

[489] For details → OSLiC, pp. 74 - 78

[490] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Execute the to-do list of use case LGPL-C2[491].
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.
- [**voluntary:**] Retain all existing copyright notices.

**prohibits** nothing explictly with respect to this use case.

### 6.9.4 LGPL-C4: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received LGPL-v2.1 resp. LGPL-v3 licensed program, application, or server (proapse) to 3rd parties – in the form of source code files or as a source code package.

**covers** OSUC-04S[492]

#### 6.9.4.1 under terms of LGPL-2.1

**requires** ... [irrelevant]

**forbids** to modify the received work in a way that the resulting "modified work" is no longer a software library (but a program)[493]. **Hence: *you are not allowed to distribute a modified program under th terms of LGPL-2.1.***

#### 6.9.4.2 under terms of LGPL-3.0

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed source code package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed,

[491] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[492] For details → OSLiC, pp. 75

[493] The LGPL-2.1 explictly requires that "the modified work must itself be a software library" (cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §2a). For details → OSLiC, p. 168

add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)

- [**mandatory:**] Give the recipient a copy of the LGPL-3.0 license. If it is not already part of the software package, add it[494].
- [**mandatory:**] Mark all modifications of source code of the program (proapse) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing LGPL licensing statements. If you add new source code files, insert a header containing your copyright line and a licensing the statement in the form required by the GNU project[495].
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications on a more functional level.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-3.0.
- [**voluntary:**] Retain all existing copyright notices.

**prohibits** nothing explictly with respect to this use case.

### 6.9.5 LGPL-C5: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received LGPL licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[496]

#### 6.9.5.1 under terms of LGPL-2.1

**requires** irrelevant

[494] For implementing the handover of files correctly → OSLiC, p. 83

[495] For details see section 'How to Apply These Terms to Your New Programs' in the GPL-v3 license.

[496] For details → OSLiC, pp. 75

**forbids** to modify the received work in a way that the resulting "modified work" is no longer a software library (but a program)[497]. **Hence: *you are not allowed to distribute a modified program under th terms of LGPL-2.1.***

#### 6.9.5.2 under terms of LGPL-3.0

**requires** to respect:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the LGPL-3.0 license. If it is not already part of the software package, add it[498].
- [**mandatory:**] Mark all modifications of source code of the program (proapse) thoroughly namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing LGPL licensing statements.
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control: Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.

[497] The LGPL-2.1 explictly requires that "the modified work must itself be a software library" (cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §2a. For details → OSLiC, pp. 168

[498] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Execute the to-do list of use case LGPL-4[499].
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications on a more functional level.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-3.0.
- [**voluntary:**] Retain all existing copyright notices.

**prohibits** nothing explictly with respect to this use case.

### 6.9.6 LGPL-C6: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received LGPL-v2.1 resp. LGPL-v3 licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[500]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-2.1 or LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed source code package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license LGPL-2.1 itself resp. by the LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the LGPL-2.1 resp. LGPL-3.0 license. If it is not already part of the software package, add it[501].

[499] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.
[500] For details → OSLiC, pp. 79
[501] For implementing the handover of files correctly → OSLiC, p. 83

- [**mandatory:**] Mark all modifications of source code of the library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing LGPL licensing statements. If you add new source code files, insert a header containing your copyright line and a licensing the statement in the form required by the GNU project[502].
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.
- [**voluntary:**] Retain all existing copyright notices.

**prohibits** ...

- to modify the library in a way that it is no longer a library (LGPL-2.1).

### 6.9.7 LGPL-C7: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received LGPL-v2.1 resp. LGPL-v3 licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[503]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-2.1 or LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the

[502] For details see section 'How to Apply These Terms to Your New Programs' in the LGPL-v2 resp. (L)GPL-v3 license.

[503] For details → OSLiC, pp. **??**

disclaimer of warranty in the form which is textually defined by the license LGPL-2.1 itself resp. by the LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)

- [**mandatory:**] Give the recipient a copy of the LGPL-2.1 resp. LGPL-3.0 license. If it is not already part of the software package, add it[504].
- [**mandatory:**] Make the source code of the distributed software accessible via a repository under your own control: Push the source code package into a repository, make it downloadable via the internet, and integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for at least 3 years after having distributed the last instance of your software package.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case LGPL-C6[505].
- [**mandatory:**] Mark all modifications of source code of the library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing LGPL licensing statements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.
- [**voluntary:**] Retain all existing copyright notices.

**prohibits** ...

- to modify the library in a way that it is no longer a library (LGPL-2.1).

[504] For implementing the handover of files correctly → OSLiC, p. 83

[505] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

### 6.9.8 LGPL-C8: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received LGPL-v2.1 resp. LGPL-v3 licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[506]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-2.1 or LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license LGPL-2.1 itself resp. by the LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)
- [**mandatory:**] Give the recipient a copy of the LGPL-2.1 resp. LGPL-3.0 license. If it is not already part of the software package, add it[507].
- [**mandatory:**] Mark all modifications of source code of the embedded library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing LGPL licensing statements. If you add new source code files to the library, insert a header containing your copyright line and a licensing the statement in the form required by the GNU project[508].
- [**mandatory:**] Maintain the structural independence of the library.
- [**mandatory:**] Let the copyright dialog of the on-top development clearly say, that it uses the LGPL licensed library. Let it reproduce the content of the existing copyright notices, a hint to the software

[506] For details → OSLiC, pp. 81

[507] For implementing the handover of files correctly → OSLiC, p. 83

[508] For details see section 'How to Apply These Terms to Your New Programs' in the LGPL-v2.1 resp. (L)GPL-v3 license.

name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.

- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also clearly say, that it uses the LGPL licensed library. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.
- [**voluntary:**] Retain all existing copyright notices.

**prohibits** ...

- to modify the library in a way that it is no longer a library (LGPL-2.1).

### 6.9.9 LGPL-C9: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received LGPL-v2.1 resp. LGPL-v3 licensed code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[509]

**requires** the tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all notices that refer to the LGPL-2.1 or LGPL-3.0 and to the absence of any warranty – are retained in your package in the form you have received them.
- [**mandatory:**] Ensure that the distributed software binary package contains a conspicuously and appropriately designed, easily to find copyright notice and a disclaimer of warranty. If these elements are missed, add a new file containing the main copyright notice and the disclaimer of warranty in the form which is textually defined by the license LGPL-2.1 itself resp. by the LGPL-3.0 itself. (Yes, repeat the disclaimer although it is also part of the license itself and although you are required to hand the license itself over to the receiver.)

[509] For details → OSLiC, pp. 81

- [**mandatory:**] Give the recipient a copy of the LGPL-2.1 resp. LGPL-3.0 license. If it is not already part of the software package, add it[510].
- [**mandatory:**] Make the source code of the embedded library and the source code of your overarching program accessible via a repository under your own control: Push the source code package into a repository and make it downloadable via the internet. Integrate an easily to find description into the distribution package which explains how the code can be received from where. Ensure, that this repository is online for as long as you continue to distribute the software.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case LGPL-C8[511].
- [**mandatory:**] Mark all modifications of source code of the embedded library (snimoli) thoroughly – namely within the source code and including the date of the modification.
- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing LGPL licensing statements.
- [**mandatory:**] Maintain the structural independence of the library.
- [**mandatory:**] Let the copyright dialog of the on-top development clearly say, that it uses the LGPL licensed library. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.
- [**mandatory:**] Either distribute the on-top development and the library in the form of dynamically linkable parts or distribute the statically linked application together with a written offer, valid for at least three years, to give the user all object-files of the on-top development and the library, so that he can relink the application on its own behalf.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a description of your modifications.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also clearly say, that it uses the LGPL licensed library. Let it reproduce the content of the existing copyright notices, a hint to the software name, a link to its homepage, the respective disclaimer of warranty, and a link to the LGPL-2.1 resp. LGPL-3.0.

[510] For implementing the handover of files correctly → OSLiC, p. 83

[511] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

- [**voluntary:**] Retain all existing copyright notices.

**prohibits** ...

- to modify the library in a way that it is no longer a library (LGPL-2.1).

### 6.9.10 Discussions and Explanations

- The LGPL-2.1 allows to "[...] to copy *and* (to) distribute verbatim copies of the Library's complete source code as you receive it [...] provided that you [a] conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; [b] keep intact all the notices that refer to this License and to the absence of any warranty; and [c] distribute a copy of this License along with the Library"[512]. Additionally, the LGPL-2.1 allows the distribution of the modfied source code "under the terms of Section 1"[513] and the distribution of binaries "under the terms of Sections 1 and 2"[514]. But the LGPL does not require any tasks if you are using the work only for yourself. Thus, the quoted conditions of "Section 1" are mandatory for all use cases concerning the distribution of an LGPL licensed work (LGPL-C2 - LGPL-C9)[515].
- Although both versions of the LGPL does not explicitly require to retain the copyright notices in the form you have received them, it is nevertheless a very good idea not to modify these elements (LGPL-C2 - LGPL-C9).
- The LGPL-2.1 allows to "[...] copy and (to) distribute the Library (or a portion or derivative of it [...]) in object code or executable form u[...] provided that you accompany it with the complete corresponding machine-readable source code [...] on a medium customarily used for software interchange". And the license subspecifies this condition in the meaning that if one makes the object code accessible without distributing it directly, then the same 'download' method for the source code fulfills this condition[516]. So, no doubt: in a very strict reading, the LGPL requires to distribute the source code and the object code together and by the same method: either both on (for example) DVD or both for being downloaded; but not the one on DVD and the other by a download repository. But the first specification also says, that the "complete corresponding machine readable source code"

[512] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §1, emphasizes by KR.
[513] cf. id., l.c., wp. §2.
[514] cf. id., l.c., wp. §4.
[515] The GPL-3.0, which is included into the LGPL-3.0, uses a similar structure to establish the same requirements (→ OSLiC, p. 151). Based on this fact one may conclude that the tasks which fulfill the corresponding LGPL-2.1 requirements together also fit the GPL-3.0 conditions and hence those of the LGPL-3.0.
[516] cf. id., ibid.

has to be distributed "on a medium customarily used for software interchange"[517]. The OSLiC understands the possibility to download files from the internet as a distribution *on a medium [being today] customarily used for software interchange*. Therefore, the OSLiC requires for all open source use cases which refer to the distribution of binaries (LGPL-C3, LGPL-C5, LGPL-C7, LGPL-C9) to make the source code of the corresponding library accessible via an internet repository[518].

- The LGPL allows to "[...] modify your copy or copies of the Library or any portion of it [...] and (to) copy and distribute such modifications [...]" only under some restrictions and condtions[519]:
  - First, modified files must be marked as modifications and marked by the date of the modification[520]. This condition must be respected by all open source use cases concerning the distribution of the modified work [LGPL-C4 - LGPL-C9], because even if one primarily intends to distribute binaries, one has also to deliver the source code. The OSLiC 'replaces' this requirement by the mandatory condition to mark each modified file and by the voluntary condition to update / generate a general changing file.
  - Second, it requires not to let the modified version depend on external data structures without "[...] (making) a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful"[521]. The OSLiC rewrites this condition as the obligation to maintain the structural independence of the library in case of using the modified library as embedded component [LGPL-C8 - LGPL-C9].
  - Third, the LGPL-2.1 definitely requires, that "the modified work must itself be a software library"[522]. This conditions can directly be incorprated as an interdiction into all use cases which refer to the modification of a library [LGPL-C6 - LGPL-C9]. But is difficult to respect this condition if one wants to modify a program which one has re-

[517] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §4.

[518] In opposite to the LGPL-2.1, the GPL-3.0, which is included into the LGPL-3.0, explictily offers the option to distribute the sources via an internet server (→ OSLiC, p. 152). So, one may again conclude that the tasks which fulfill the corresponding LGPL-2.1 requirements together also fit the GPL-3.0 and the LGPL-3.0 conditions.

[519] cf. id., l.c., wp. §2.

[520] For LGPL-2.1 see cf. id., ibid.. For GPL-3.0 (being included into the LGPL-3.0) see cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §5

[521] For LGPL-2.1 see cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §2d. For LGPL-3.0 see cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp. §2a

[522] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §2.

ceived under the terms of the LGPL-2.1. Logically, one can write an application and license it under the LGPL. But - as a consequence - that impedes the modification of this work because the result must be a library. The LGPL-3.0 does not contain such a requirement. Hence, the OSLiC allows the distribution of modified programs (LGPL-C4, LGPL-C5) only if they are licensed under the terms of LGPL-3.0as long as nobody has shown us an exit out of this trap..

- Additionally, the LGPL-2.1 allows the licensee to distribute an overarching on-top development – in the wording of the LGPL-2.1: a "work that uses the libary"[523] – "as an exception to the Sections above" in "combination" with the library "under terms of your choice"[524], provided that the licensee fulfills additional conditions: First, it must clearly be stated that the on-top development depends on the (modified) library. Second, the LGPL must be added into the distributed package. Third, in its own copyright dialog, the on-top development must mention the library, its copyright holder, and that it is licensed under the LGPL[525]. In the LGPL-3.0, this condition is similarily integrated: On the one hand, the "combined work" is defined as "a work produced by combining or linking an Application with the Library"[526]. On the other hand, the LGPL-3.0 states that one "[...] may convey a Combined Work under terms of (his own) choice" provided that one [a] clearly says that the overarching on-top development uses the LGPL licensed library, [b] distributes the LGPL-3.0 and the GPL-3.0 license as part of the package, [c] displays all these (licensing) information by the existing display technologies, [d] requires an appropriate shared library mechanism, and [e] offers the respectiv installion information[527]. These requirements can directly be inserted as conditions into the respective use cases – namely for both LGPL versions (LGPL-C8, LGPL-C9).
- The most difficult requirements of the LGPL-2.1 concern the distribution in the form of binaries. In a very strict reading, the LGPL does not require to link the on-top development and the libary only dynamically. At first, the LGPL mentions, that the "[..] work (that uses the Library), in isolation, is not a derivative work of the Library [...]". But if it is linked to the library the resulting executable program – of course – becomes "a derivative of the Library" and that it is therefore "[...] covered by this License (LGPL-2.1)". But the LGPL-2.1 directly continues this statement with the hint, that "Section 6 states terms for distribution of such executables"[528]. Finally, this

[523] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §5, §6.
[524] cf. id., l.c., wp. §6.
[525] cf. id., ibid.
[526] cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp. §0.
[527] cf. id., l.c., wp. §4.
[528] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §5.

section 6 directly starts with statement: "As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice"[529].

This is important to know, because until this section 6 one can not directly read or indirectly infer that the LGPL-2.1 distinguished the act of dynamically linking a program and a lirbrary from that of statically linking these parts. The LGPL only wants to ensure that the binaries of the library itself can be replaced by a newer version. And that is required by section 6[530]. From a technically viewpoint, this can only be guaranteed, if the binaries of the on-top development and the library together are "used in a suitable shared library mechanism"[531] or if one also gets all compiled, but not linked object-files of the on-top development and the library, either directly, or via using a "a written offer, valid for at least three years, to give the same user the (respective) materials"[532]. In the first case, the user can replace the received version of the library and can let the application be relinked automatically. In the second case, he has to do it manually. It is important to know that these ways exist if one wants or must distribute statically linked works. The LGPL-2.1 does not forbid to distribute statically linked applications. But it requires to enable the receiver to relink the work.

The LGPL-3.0 has reduced these complex conditions in a special way: First, it does not use the words 'statically linked' or 'dynamically' linked. Second it defines the combined work 'only' as the result of "combining or linking an Application with the Library"[533]. But then it requires for the distribution of the combined works that one has either to "convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work [...]" or that one must presuppose that the receiver uses "[...] suitable shared library mechanism for linking with the Library [...] that [...] operate properly with a modified version of the Library [...]"[534]. Finally, the LGPL-3.0 adds that in the first case the these materials which enables the relinking must be distributed "[...] in the manner specified by section 6 of the GNU GPL[-3.0] for conveying Corresponding Source"[535]. And this section 6 of

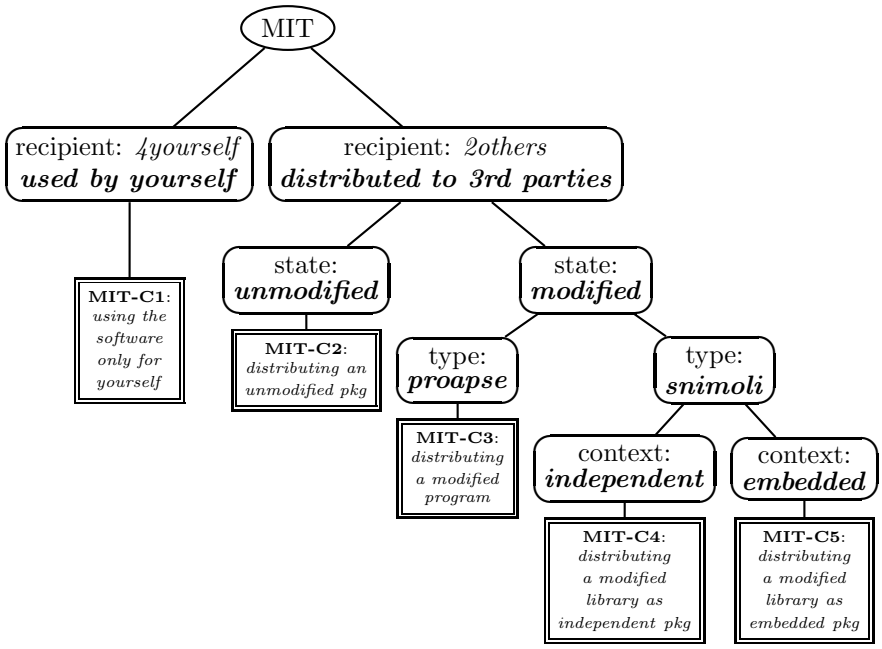
529) cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §6.
530) cf. id., ibid.
531) cf. id., ibid.
532) cf. id., ibid.
533) cf. *Open Source Initiative*: The LGPL-3.0 License (OSI), 2007, wp. §0.
534) cf. id., l.c., wp. §4.
535) cf. id., ibid.

the GPL-3.0 allows the well known method to "convey the object code [...] accompanied by a written offer [...] to give anyone [...] access to copy the Corresponding Source from a network server at no charge"[536]

Therefore, the OSLiC can condense these conditions into the requirement, either to distribute dynamically linkable parts, or to distribute statically linked applications "(accompanied) [...] with a written offer, valid for at least three years, to give the same user the [complete] materials"[537], so that he can relink the application on its own behalf. It is clear, that this condition is only valid for the use case LGPL-C9.

## 6.10 MIT licensed software

The MIT license is known as one of the most permissive licenses. Thus, the MIT specific finder can be simplified:




### 6.10.1 MIT-C1: Using the software only for yourself

**means** that you are going to use a received MIT software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[538]

[536] cf. *Open Source Initiative*: The GPL-3.0 License (OSI), 2007, wp. §6.
[537] cf. *Open Source Initiative*: The LGPL-2.1 License (OSI), 1999, wp. §6.
[538] For details → OSLiC, pp. 73 - 80

**requires** no tasks in order to fulfill the conditions of the MIT license with respect to this use case:

- You are allowed to use any kind of MIT licensed software in any sense and in any context without any other obligations if you do not handover the software to 3rd parties and if you do not modify the existing copyright notes and the existing permission notice.

**prohibits** nothing explicitly.

### 6.10.2 MIT-C2: Passing the unmodified software

**means** that you are going to distribute an unmodified version of the received MIT software to 3rd parties – in the form of binaries or as source code files. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent package.

**covers** OSUC-02S, OSUC-02B, OSUC-05S, OSUC-05B, OSUC-07S, OSUC-07B[539]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. the MIT license text containing the specific copyright notices of the original author(s), the permission notices and the MIT disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage.

**prohibits** nothing explicitly.

### 6.10.3 MIT-C3: Passing a modified program

**means** that you are going to distribute a modified version of the received MIT program, application, or server (proapse) to 3rd parties – in the form of binaries or as source code files.

**covers** OSUC-04S, OSUC-04B[540]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the original licensing elements – esp. the MIT license text containing the specific copyright notices of the ori-

[539] For details → OSLiC, pp. 73 - 78
[540] For details → OSLiC, pp. 75

ginal author(s), the permission notices and the MIT disclaimer – are retained in your package in the form you have received them.

- [**voluntary:**] Mark your modifications in the source code, regardless whether you want to distribute the code or not.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage.
- [**voluntary:**] You can expand an existing copyright notice presented by the program with information about your own work or modifications.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the program also state that it is based on a version originally licensed under the MIT license. Because you are already modifying the program, you can also add such a hint, if the presented original copyright notice lacks such a statement.

**prohibits** nothing explicitly.

### 6.10.4 MIT-C4: Passing a modified library independently

**means** that you are going to distribute a modified version of the received MIT code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binaries or as source code files. But you do not embed it into another larger software unit.

**covers** OSUC-08S, OSUC-08B[541]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the original licensing elements – esp. the MIT license text containing the specific copyright notices of the original author(s), the permission notices and the MIT disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Mark your modifications in the source code, regardless whether you want to distribute the code or not.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage.

**prohibits** nothing explicitly.

[541] For details → OSLiC, pp. 79

### 6.10.5 MIT-C5: Passing a modified library as embedded component

**means** that you are going to distribute a modified version of the received MIT code snippet, module, library, or plugin (snimoli) to 3rd parties together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component – regardless whether you distribute it in the form of binaries or as source code files.

**covers** OSUC-10S, OSUC-10B[542]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the original licensing elements – esp. the MIT license text containing the specific copyright notices of the original author(s), the permission notices and the MIT disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Mark your modifications in the source code, regardless whether you want to distribute the code or not.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the program uses a component being licensed under the MIT license. And it is a good tradition to insert links to the homepage / download page of this used component.
- [**voluntary:**] It's also a good tradition to let the documentation of your program and/or your additional material also mention that you have used this component added by a link to the original software component and its homepage.
- [**voluntary:**] Arrange your distribution so that the original licensing elements – esp. the MIT license text containing the specific copyright notices of the original author(s), the permission notices and the MIT disclaimer – clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the libraries, modules, snippet, or plugins in specific directories which contain also all licensing elements.

**prohibits** nothing explicitly.

### 6.10.6 Discussions and Explanations

The MIT-License is known as one of the most permissive licenses. It is a very short license containing (0) a copyright notice, (1) a paragraph saying that you

[542] For details → OSLiC, pp. 81

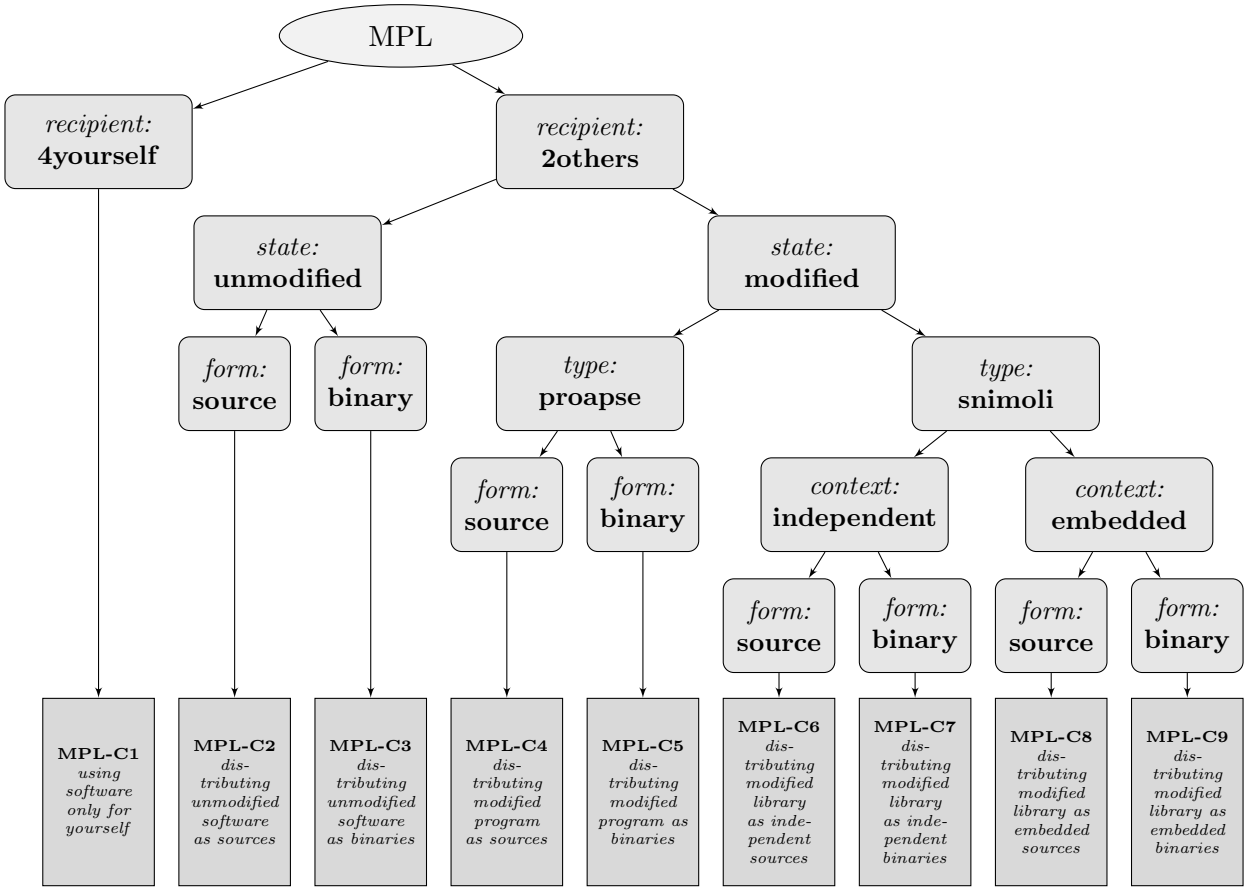
are allowed to do almost anything you want, followed (2) by the condition that you have to "include" the existing copyright notes and the permission notes "[...] in all copies or substantial portions of the software", and (3) closed by the well known disclaimer[543]. But the license doesn't talk about the difference of source code and object code. So, you have to find the right way by yourself. Here are our readings:

- If you do not modify the received MIT licensed application, neither for your own purposes, nor for handing over the program to 3rd parties, you can conclude that all copyright notices and permission notices are already correct.
- Nevertheless, we added the hint not to modify these licensing elements in the context of the use case *used by yourself*. This is evoked by the MIT license itself. It requires explicitly that " the above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software"[544] – thus also into those copies you make for your own purposes own your own machines, and even if this is probably not so often reviewed.
- If you modify the received MIT licensed application, regardless for which purposes, you are simply not allowed to erase or modify existing copyright notes and permission notices. You may add your own modifications under new conditions, but the old base must survive.

## 6.11 MPL licensed software

Also, the Mozilla Public License clearly distinguishes the distribution in the form of source code from that in the form of binaries: First, it allows the "Distribution of Source Form"[545]. Then, it specifies the conditions for a "Distribution of Executable Form"[546]. Additionally, the MPL confronts the "distribution of Covered Software" with the "distribution of a Larger Work"[547]. So, taken as whole, the MPL mainly focusses on the distribution of software. Thus, for finding the relevant, simply processable task lists, also the following MPL specific open source use case structure[548] can be used:

[543] cf. *Open Source Initiative*: The MIT License, 2012, wp.
[544] cf. id., ibid.
[545] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §3.1.
[546] cf. id., l.c., wp. §3.2.
[547] cf. id., l.c., wp. §3.3.
[548] For details of the general OSUC finder → OSLiC, pp. 66 and 69




### 6.11.1 MPL-C1: Using the software only for yourself

**means** that you are going to use a received MPL licensed software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[549]

**requires** no tasks in order to fulfill the conditions of the MPL 2.0 license with respect to this use case:

- You are allowed to use any kind of MPL software in any sense and in any context without being obliged to do anything as long as you do not give the software to 3rd parties.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.

[549] For details → OSLiC, pp. 73 - 80

- to promote any of your services – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.11.2 MPL-C2: Passing the unmodified software as source code

**means** that you are going to distribute an unmodified version of the received MPL software to 3rd parties - in the form of source code files or as a source code package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit

**covers** OSUC-02S, OSUC-05S, OSUC-07S[550]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Give the recipient a copy of the MPL 2.0 license. If it is not already part of the software package, add it[551]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the MPL 2.0 license.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.
- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, ex-

[550] For details → OSLiC, pp. 73 - 78
[551] For implementing the handover of files correctly → OSLiC, p. 83

cept as required for unpartially describing the used software and reproducing the copyright notice.

### 6.11.3 MPL-C3: Passing the unmodified software as binaries

**means** that you are going to distribute an unmodified version of the received MPL software to 3rd parties – in the form of binary files or as a binary package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02B, OSUC-05B, OSUC-07B[552]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] Make the source code of the software accessible via a repository under your own control: Push the source code package into an internet repository and enable its download function without requiring any fee from the downloading user. Integrate an easily to find description into your distribution package which explains how the code can be received from where. Ensure, that this repository is usable reasonably long enough.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case MPL-C2[553].
- [**voluntary:**] Give the recipient a copy of the MPL 2.0 license. If it is not already part of the software package, add it[554]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*

[552] For details → OSLiC, pp. 74 - 78

[553] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[554] For implementing the handover of files correctly → OSLiC, p. 83

- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the MPL 2.0 license.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.
- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.11.4 MPL-C4: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received MPL licensed program, application, or server (proapse) to 3rd parties – in the form of source code files or a source code package.

**covers** OSUC-04S[555]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Give the recipient a copy of the MPL 2.0 license. If it is not already part of the software package, add it[556]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing MPL licensing statements. If you add new source code files, insert a header containing your copyright line and an MPL adequate licensing the statement.

[555] For details → OSLiC, pp. 75
[556] For implementing the handover of files correctly → OSLiC, p. 83

- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**voluntary:**] Mark all modifications of the source code of the program (proapse) thoroughly – namely within the modfied source code.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the MPL 2.0 license.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.
- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.11.5 MPL-C5: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received MPL licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[557]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] Make the source code of the software accessible via a repository under your own control: Push the source code package into an internet repository and enable its download function without requiring any fee from the downloading user. Integrate an easily to find description into your distribution package which explains how the

[557] For details → OSLiC, pp. 75

code can be received from where. Ensure, that this repository is usable reasonably long enough.

- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case MPL-C4[558].
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing MPL licensing statements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**voluntary:**] Give the recipient a copy of the MPL 2.0 license. If it is not already part of the software package, add it[559]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the MPL 2.0 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.
- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

[558] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[559] For implementing the handover of files correctly → OSLiC, p. 83

### 6.11.6 MPL-C6: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received MPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

**covers** OSUC-08S[560]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Give the recipient a copy of the MPL 2.0 license. If it is not already part of the software package, add it[561]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing MPL licensing statements. If you add new source code files, insert a header containing your copyright line and an MPL adequate licensing the statement.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**voluntary:**] Mark all modifications of the source code of the library (snimoli) thoroughly – namely within the modfied source code.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the MPL 2.0 license.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices,

[560] For details → OSLiC, pp. 79
[561] For implementing the handover of files correctly → OSLiC, p. 83

patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.

- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.11.7 MPL-C7: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received MPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[562]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] Make the source code of the software accessible via a repository under your own control: Push the source code package into an internet repository and enable its download function without requiring any fee from the downloading user. Integrate an easily to find description into your distribution package which explains how the code can be received from where. Ensure, that this repository is usable reasonably long enough.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case MPL-6[563].
- [**mandatory:**] Organize your modifications in a way that they are covered by the existing MPL licensing statements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution.

[562] For details → OSLiC, pp. 79

[563] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

- [**voluntary:**] Give the recipient a copy of the MPL 2.0 license. If it is not already part of the software package, add it[564]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the software name, a link to its homepage, and a link to the MPL 2.0 license – especially as a subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.
- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.11.8 MPL-C8: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received MPL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[565]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them.
- [**mandatory:**] Give the recipient a copy of the MPL 2.0 license. If it

[564] For implementing the handover of files correctly → OSLiC, p. 83
[565] For details → OSLiC, pp. 81

is not already part of the software package, add it[566]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*

- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing MPL licensing statements. If you add new source code files to the library itself, insert a header containing your copyright line and an MPL adequate licensing the statement.
- [**voluntary:**] Arrange your source code distribution so that the integrated MPL and the *licensing files* clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the embedded components like libraries, modules, snippets, or plugins in specific directory which contains also all additional licensing elements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**voluntary:**] Mark all modifications of the source code of the embedded library (snimoli) thoroughly – namely within the source code.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the name of the used MPL licensed component, a link to its homepage, and a link to the MPL 2.0 license – especially as subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.
- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

[566] For implementing the handover of files correctly → OSLiC, p. 83

### 6.11.9 MPL-C9: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received MPL licensed code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[567]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the licensing elements – esp. all copyright notices, patent notices, disclaimers of warranty, or limitations of liability – are retained in your package in exact the form you have received them. If you compile the binary from the sources, ensure that all these licensing elements are also incorporated into the package.
- [**mandatory:**] Make the source code of the embedded library accessible via a repository under your own control: Push the source code package into an internet repository and enable its download function without requiring any fee from the downloading user. Integrate an easily to find description into your distribution package which explains how the code can be received from where. Ensure, that this repository is usable reasonably long enough.
- [**mandatory:**] Insert a prominent hint to the download repository into your distribution and/or your additional material.
- [**mandatory:**] Execute the to-do list of use case MPL-C8[568].
- [**mandatory:**] Organize your modifications of the embedded library in a way that they are covered by the existing MPL licensing statements.
- [**voluntary:**] Create a *modification text file*, if such a notice file still does not exist. *Expand* the *modification text file* by a more general description of your modifications. Incorporate it into your distribution package.
- [**voluntary:**] Give the recipient a copy of the MPL 2.0 license. If it is not already part of the software package, add it[569]. If the licensing statement in the licensing file of the package does still not clearly state that the package is licensed under the MPL, additionally insert your own correct MPL licensing file containing the sentence: *This Source*

[567] For details → OSLiC, pp. 81

[568] Making the code accessible via a repository means distributing the software in the form of source code. Hence, you must also fulfill all tasks of the corresponding use case.

[569] For implementing the handover of files correctly → OSLiC, p. 83

*Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.*

- [**voluntary:**] Arrange your binary distribution so that the integrated MPL and the *licensing files* clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the embedded components like libraries, modules, snippets, or plugins in specific directory which contains also all additional licensing elements.
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also reproduce the content of the existing *copyright notice text files*, a hint to the name of the used MPL licensed component, a link to its homepage, and a link to the MPL 2.0 license – especially as subsection of your own copyright notice.

**prohibits** ...

- to remove or to alter any license notices – including copyright notices, patent notices, disclaimers of warranty, or limitations of liablility – contained within the software package you have received.
- to promote any of your products – based on the this software – by trademarks, service marks, or logos linked to this MPL software, except as required for unpartially describing the used software and reproducing the copyright notice.

### 6.11.10 Discussions and Explanations

The MPL offers a section "Responsibilities" which contains nearly all requirements[570]. Only for some subordinated aspects, one has also to reflect other paragraphs[571]. With respect to this structure, we can detect the following tasks:

- In a more general attitude, the MPL states that it "[...] does not grant any rights in the trademarks, service marks, or logos of any Contributor" – except as it may be necessary "to comply with" other requirements of the license[572]. The OSLiC rewrites the message as the interdiction to promote own services and products by and with such elements.
- The MPL also generally prescribes that "you may not remove or alter the substance of any license notice (including copyright notices, patent notices, disclaimer of warranties, or limitations of liabiliy) contained within the

[570] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §3.
[571] pars pro to cf. id., l.c., wp. §3 - concerning the trademarks.
[572] cf. id., l.c., wp. §2.3.

Source Code Form [...]"[573]. This focussing to the "substance of any license notice" refers to the allowance to "[...] alter any license notices to the extent required to remedy known factual innacuracies"[574]. Following its principle to offer one reliable way and to ignore variants of secondary importance, the OSLiC simplifies this condition to the general proscription to modify any licensing material – namely for all use cases [MPL-C1 - MPL-C9]. But for emphasizing that this is a job which must be activily done, the OSLiC additionally rewrites this interdiction into all *2others* use cases [MPL-C2 - MPL-C9] as the task to retain the licensing notices in the form one has obtained them.

- Moreover, the MPL requires for all "distributions of [the] source [code] form" that all modifications of the software "[...] must be under the terms of (the MPL)" and that the distributor "[...] must inform" all "recipients" that the software "[...] is governed by the terms of (the MPL), and how (the recipients) can obtain a copy of this license"[575]. For the respective use case (MPL-C2, MPL-C4, MPL-C6, MPL-C8), the OSLiC rewrites these conditions so that each MPL source code package must mandatorily contain the MPL itself as textfile and an additional licensing file or statement strictly following the text given by the addendum of the MPL[576]. Because the MPL is 'only' a license with weak copyleft, the OSLiC proposes to separate the MPL licensed, embedded component from the overarching program (MPL-C8).
- But the MPL does not explicitly require to mark all modifications. Nevertheless, this is state of the art in computer emgineering. Therefore, with respect to the cases of distributing modified source code (MPL-C4, MPL-C6 and MPL-C8), the OSLiC proposes to mark all modifications inside of the source code and to update the description of the functional changes. In case of distributing the modified software in the form of binaries, it should be sufficient, to describe the modifications only on the functional level.
- Furthermore, the MPL requires that the "Covered Software" – in all cases of distributing it in an "Executable Form" (MPL-C3, MPL-C5, MPL-C7, MPL-C9) – "[...] must also be made available in Source Code Form [...]" and that the distributor "[...] must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient"[577]. The OSLiC rewrites these conditions as the obligation to offer a download service at no charge and to point towards this services

[573] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §3.4.
[574] cf. id., ibid.
[575] cf. id., l.c., wp. §3.1.
[576] cf. id., l.c., wp. Exhibit A.
[577] cf. id., l.c., wp. §3.2.a.

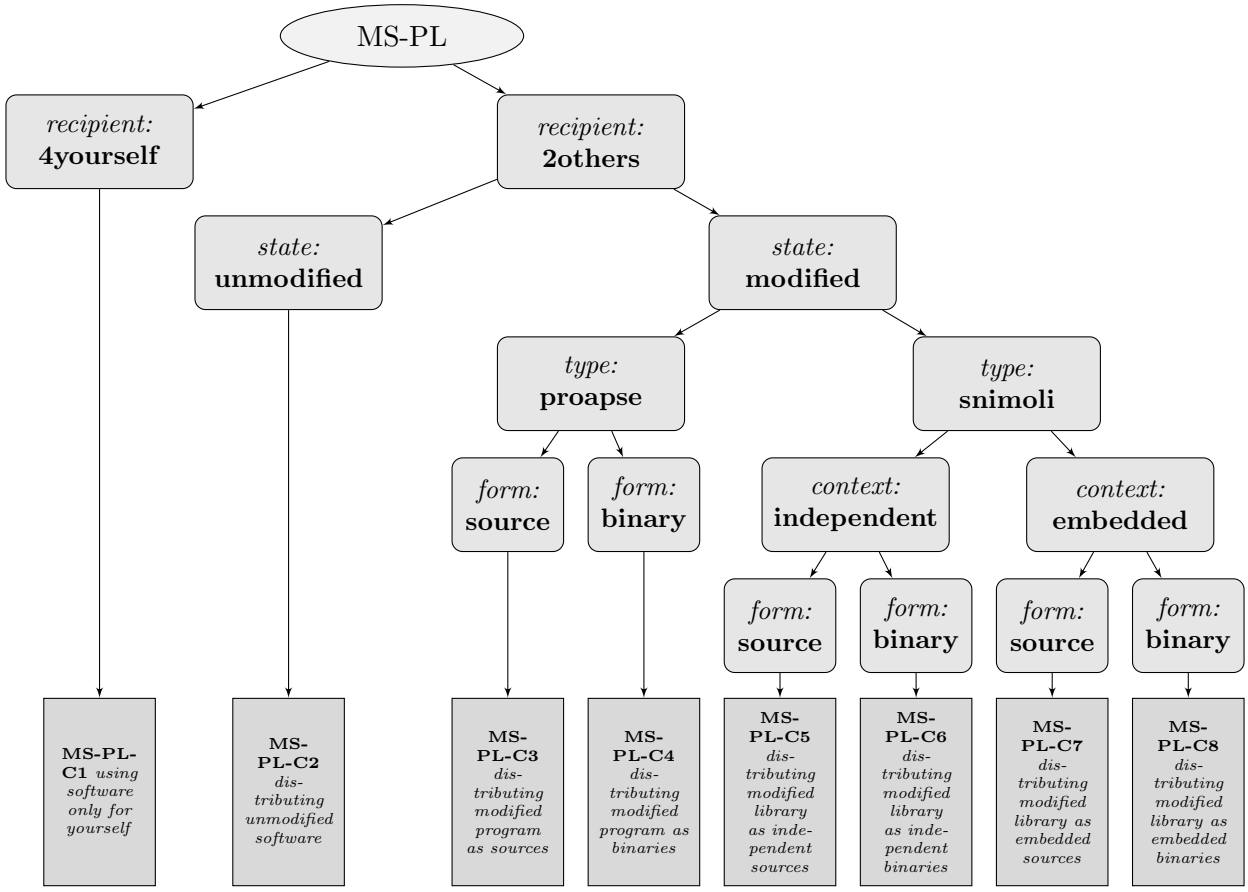
inside of the distributed package.

- In this context, the MPL allows to distribute the binaries under terms of another license "[...] provided that that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License"[578]. This possibility might become important for those cases where the license compatibility must explicitly be managed. Normally, it should be sufficient also to distribute the binaries under the MPL. Thus, in case of distributing binaries ((MPL-C3, MPL-C5, MPL-C7, MPL-C9), the OSLiC proposes to insert into the distribution packages the MPL itself and an additional licensing file or statement strictly following the text given by the addendum of the MPL[579]. But again, because the MPL is 'only' a license with weak copyleft, the OSLiC proposes to separate the MPL licensed embedded component from the overarching program (MPL-C9)
- Finally, one has clearly to state that this rule above evokes a real source code distribution which therefore must follow the rules of distributing the software. Thus, the OSLiC requires in all cases of a binary distribution to execute also the task-lists of the respective source code use cases.

## 6.12 Microsoft Public License

The MS-PL license is also one of the most permissive licenses. Thus, the MS-PL specific finder can be simplified:

[578] cf. *Open Source Initiative*: The MPL-2.0 License (OSI), 2013, wp. §3.2.b.
[579] cf. id., l.c., wp. Exhibit A.




### 6.12.1 MS-PL-C1: Using the software only for yourself

**means** that you are going to use a received MS-PL software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[580]

**requires** no tasks in order to fulfill the conditions of the MS-PL license with respect to this use case:

- You are allowed to use any kind of MS-PL licensed software in any sense and in any context without any other obligations if you do not handover the software to 3rd parties.

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval).

[580] For details see pp. 73 - 80

### 6.12.2 MS-PL-C2: Passing the unmodified software

**means** that you are going to distribute an unmodified version of the received MS-PL software to 3rd parties – in the form of binaries or as source code files. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent package.

**covers** OSUC-02S, OSUC-02B, OSUC-05S, OSUC-05B, OSUC-07S, OSUC-07B[581]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that all licensing elements – esp. all copyright, patent, trademark, and attribution notices that are part of the version you received – are completely retained in your package.
- [**mandatory:**] Incorporate a complete copy of the MS-PL license into your package, regardless whether you distribute a source code or a binary package[582].
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage.

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval).

### 6.12.3 MS-PL-C3: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received MS-PL licensed program, application, or server (proapse) to 3rd parties - in the form of source code files or source code package.

**covers** OSUC-04S[583]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that all licensing elements – esp. all copyright, patent, trademark, and attribution notices that are part of the version you received – are completely retained in your package.
- [**mandatory:**] Incorporate a complete copy of the MS-PL license into your package.

[581] For details → OSLiC, pp. 74 - 78
[582] → OSLiC, p. 196
[583] For details → OSLiC, pp. 75

- [**mandatory:**] If you do not want to publish your modifications under the MS-PL too, then cleanly separate your own sources and licensing documents from original elements of the adopted work.
- [**voluntary:**] Mark your modifications in the sourcecode.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage (as far as this does not clashes with the prohibitions stated below).
- [**voluntary:**] You are allowed to expand an existing copyright notice of the program to mention your own contributions.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the program is licensed under the MS-PL license (as far as this does not clashes with the prohibitions stated below). Because you are already modifying the program, you can also add such a hint, if the original copyright notice lacks such a statement.

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval).

### 6.12.4 MS-PL-C4: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received MS-PL licensed program, application, or server (proapse) to 3rd parties – in the form of binary files or as bianry package.

**covers** OSUC-04B[584]

**requires** the following tasks in order to fulfill the license conditions:

- [**voluntary:**] Mark your modifications in the source code even if you do not intend to distribute it.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage (as far as this does not clashes with with the prohibitions stated below).
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the derivative work is based on a version originally licensed under the MS-PL license (as far as this does not clashes with the prohibitions

[584] For details → OSLiC, pp. 75

stated below) – perhaps by linking to the project homepage of the original. Because you are already modifying the program, you can also add such a hint, if the original copyright notice lacks such a statement.

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval)

### 6.12.5 MS-PL-C5: Passing a modified library independently as source code

**means** that you are going to distribute a modified version of the received MS-PL code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code but without embedding it into another larger software unit.

**covers** OSUC-08S[585]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that all licensing elements – esp. all copyright, patent, trademark, and attribution notices that are part of the version you received – are completely retained in your package.
- [**mandatory:**] Incorporate a complete copy of the MS-PL license into your package.
- [**mandatory:**] If you do not want to publish your modifications under the MS-PL too, then cleanly separate your own sources and licensing documents from original elements of the adopted part(s).
- [**voluntary:**] Mark your modifications in the sourcecode.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage (as far as this does not clashes with with the prohibitions stated below).

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval)

### 6.12.6 MS-PL-C6: Passing a modified library independently as binary

**means** that you are going to distribute a modified version of the received MS-PL code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files but without embedding it into another larger software unit.

[585] For details → OSLiC, pp. 79

**covers** OSUC-08B[586]

**requires** the following tasks in order to fulfill the license conditions:

- [**voluntary:**] Mark your modifications in the source code even if do not want to distribute it.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage (as far as this does not clashes with with the prohibitions stated below).

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval)

### 6.12.7 MS-PL-C7: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received MS-PL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[587]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that all licensing elements – esp. all copyright, patent, trademark, and attribution notices that are part of the version you received – are completely retained in your package.
- [**mandatory:**] Incorporate a complete copy of the MS-PL license into your package.
- [**mandatory:**] If you do not want to publish your modifications and/or your overarching application under the MS-PL too, then cleanly separate your own sources and licensing documents from original elements of the adopted work.
- [**voluntary:**] Mark your modifications in the sourcecode.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage (as far as this does not clashes with with the prohibitions stated below).

[586] For details → OSLiC, pp. 79
[587] For details → OSLiC, pp. 81

- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice shown by your overarching program also state that it is based on a component originally licensed under the MS-PL license – perhaps by linking the project homepage of the original (as far as this does not clashes with the prohibitions stated below).

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval)

### 6.12.8 MS-PL-C8: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received MS-PL licensed code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[588]

**requires** the following tasks in order to fulfill the license conditions:

- [**voluntary:**] Mark your modifications in the source code even if do not want to distribute it.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage (as far as this does not clashes with with the prohibitions stated below).
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice shown by your own overarching program also state that it is based on a component originally licensed under the MS-PL license – perhaps by linking the project homepage of the original (as far as this does not clashes with the prohibitions stated below).

**prohibits** to use any contributors' name, logo, or trademarks (without an additional or general legally based approval)

### 6.12.9 Discussions and Explanations

The MS-PL is also a very permissive and short license. It requires to do: (a) You must preserve existing licensing elements. (b) You must distribute the source code as whole or "portions" of the source code under the MS-PL. (c) You must add a copy of the license if you distribute (parts of) the source code. (d) If you

[588] For details → OSLiC, pp. 81

distribute a binary package, you must distribute (the parts of) the work under a license "that complies with this (MS-PL) license"[589].

The most confusing clause is probably the condition, to "[...] distribute any portion of the software in compiled or object code form [...] only [...] under a license that complies with this license". But a closer examination is lighting the situation: The only other conditions of the license which refer to the context of distributing binaries are the requirements a) not to abuse trademarks, b) not to bring a patent claim against any contributor, and c) not to expect any warranties or guarantees with respect to the distributed portion[590].

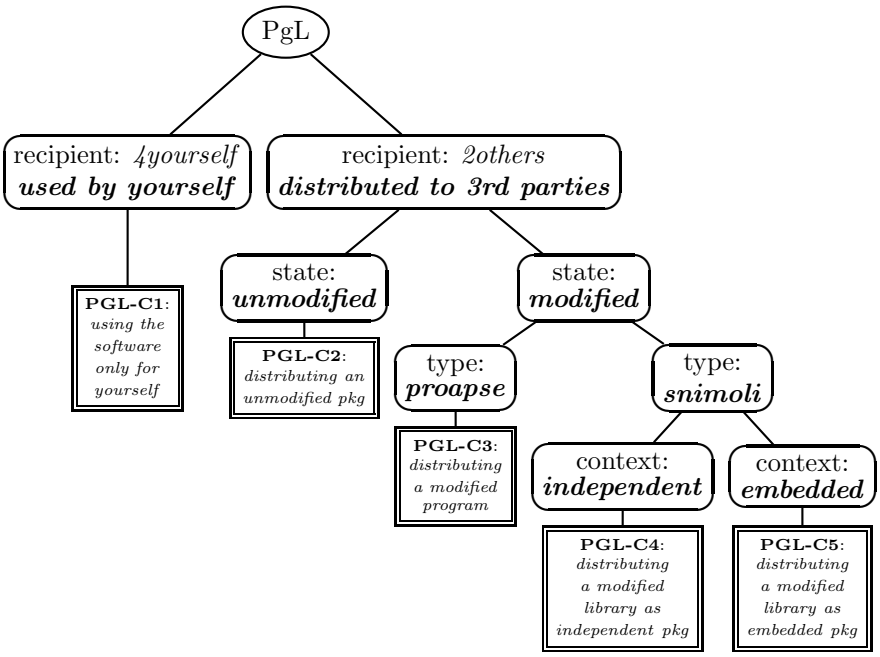
Based on these readings we decided ...

- ...to let you incorporate a copy of the license into your distribution even if it only contains the binaries of the unmodified version: if you have not modified it, you do not lose any advantage if you add the license, too. So, this is the best method to fulfill the *MSL-PL binary condition*.
- ...to erase all mandatory conditions in case of the binary distributions: the patent restriction of the MS-PL itself is already covered by the MS-PL patent section of the OSLiC[591] and the no warranty clause of the MS-PL by the OSLiC section concerning the power of the MS-PL[592] while the trademark restrictions are explicitly added into the prohibition section.
- ...to erase the hints to a voluntarily updated copyright dialog in case of distributing a snimoli independently because the copyright dialog normally is designed by the overarching work which uses the library, not by the library itself.

## 6.13 Postgres Licensed Software

Like the MIT License Postgres License is a very permissive licenses. Thus, the PGL specific finder can be simplified:

589) cf. *Open Source Initiative*: MS-PL, 2013, wp.
590) cf. id., l.c., wp. §3A, §3B, §3E.
591) → OSLiC, p. 57
592) → OSLiC, p. 43




### 6.13.1 PGL-C1: Using the software only for yourself

**means** that you are going to use a received PGL software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[593]

**requires** no tasks in order to fulfill the conditions of the PGL with respect to this use case:

- You are allowed to use any kind of PGL licensed software in any sense and in any context without any other obligations if you do not handover the software to 3rd parties and if you do not modify the existing copyright notes and the existing permission notice.

**prohibits** nothing explicitly.

### 6.13.2 PGL-C2: Passing the unmodified software

**means** that you are going to distribute an unmodified version of the received PGL software to 3rd parties – regardless whether you distribute it in the form of binaries or as source code files. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent package.

[593] For details → OSLiC, pp. 73 - 80

**covers** OSUC-02S, OSUC-02B, OSUC-05S, OSUC-05B, OSUC-07S, OSUC-07B[594]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete Postgres License including the copyright notice, the permission notices, and the PGL disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage.

**prohibits** nothing explicitly.

### 6.13.3 PGL-C3: Passing a modified program

**means** that you are going to distribute a modified version of the received PGL program, application, or server (proapse) to 3rd parties – regardless whether you distribute it in the form of binaries or as source code files.

**covers** OSUC-04S, OSUC-04B,[595]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete Postgres License including the copyright notice, the permission notices, and the PGL disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Mark your modifications in the source code, regardless whether you want to distribute the code or not.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage.
- [**voluntary:**] You can expand an existing copyright notice presented by the program with information about your own work or modifications.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the program also state that it is based on a version originally licensed under the PGL license. Because you are already modifying the program, you can also add such a hint, if the presented original copyright notice lacks such a statement.

**prohibits** nothing explicitly.

[594] For details → OSLiC, pp. 73 - 78
[595] For details → OSLiC, pp. 75

### 6.13.4 PGL-C4: Passing a modified library independently

**means** that you are going to distribute a modified version of the received PGL code snippet, module, library, or plugin (snimoli) to 3rd parties without embedding it into another larger software unit – regardless whether you distribute it in the form of binaries or as source code files.

**covers** OSUC-08S, OSUC-08B[596]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete Postgres License including the copyright notice, the permission notices, and the PGL disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Mark your modifications in the source code, regardless whether you want to distribute the code or not.
- [**voluntary:**] It's a good tradition to let the documentation of your distribution and/or your additional material also contain a link to the original software (project) and its homepage.

**prohibits** nothing explicitly.

### 6.13.5 PGL-C5: Passing a modified library as embedded component

**means** that you are going to distribute a modified version of the received PGL code snippet, module, library, or plugin (snimoli) to 3rd parties together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component – regardless whether you distribute it in the form of binaries or as source code files.

**covers** OSUC-10S, OSUC-10B[597]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete Postgres License including the copyright notice, the permission notices, and the PGL disclaimer – are retained in your package in the form you have received them.
- [**voluntary:**] Mark your modifications in the source code, regardless whether you want to distribute the code or not.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the program uses a component being licensed under the PGL

[596] For details → OSLiC, pp. 79
[597] For details → OSLiC, pp. 81

license. And it is a good tradition to insert links to the homepage / download page of this used component.

- [**voluntary:**] It's also a good tradition to let the documentation of your program and/or your additional material also mention that you have used this component added by a link to the original software component and its homepage.
- [**voluntary:**] Arrange your distribution so that the original licensing elements – esp. the PGL license text containing the specific copyright notices of the original author(s), the permission notices and the PGL disclaimer – clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the libraries, modules, snippet, or plugins in specific directories which contain also all licensing elements.

**prohibits** nothing explicitly.

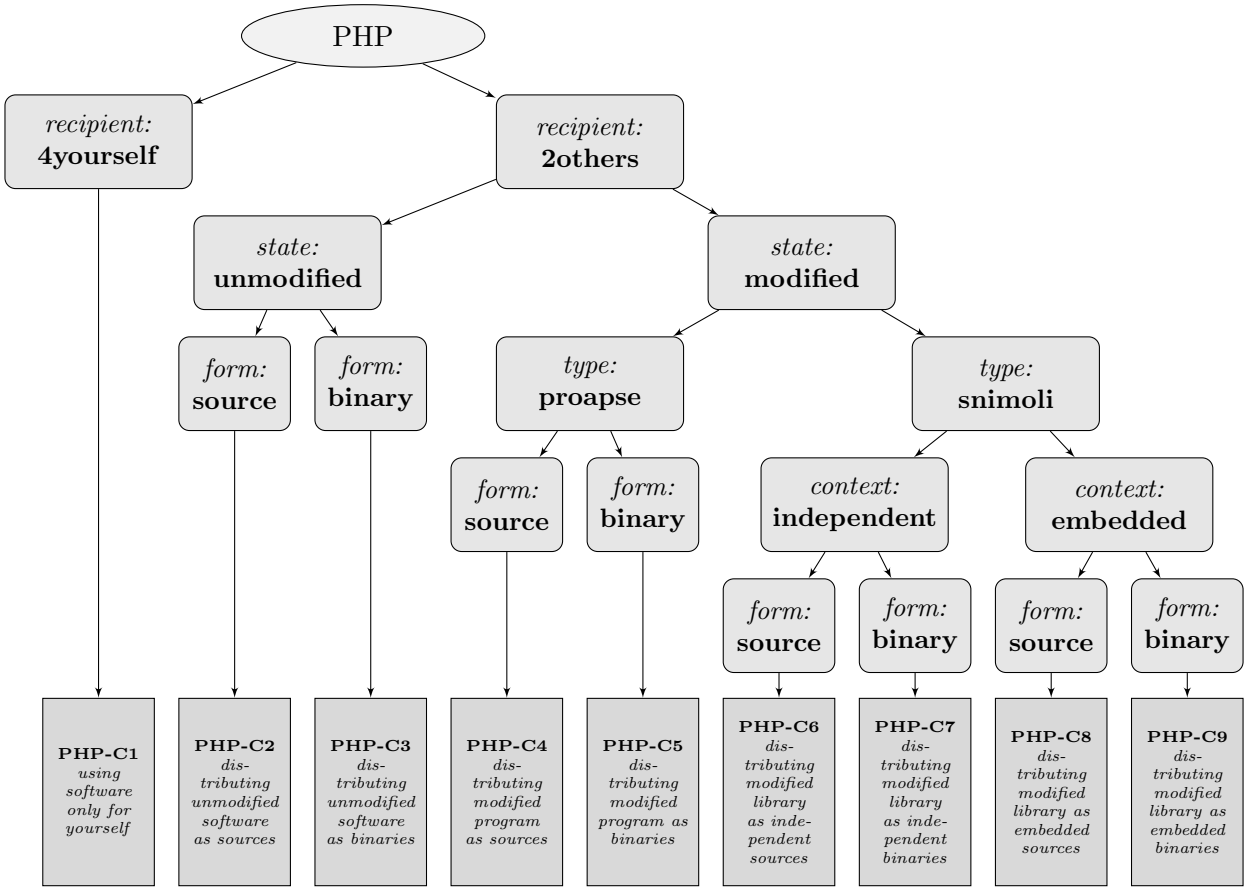
### 6.13.6 Discussions and Explanations

The PGL-License follows the structure of the MIT license: it also contains (1) a copyright notice, (2) a paragraph saying that you are allowed to do almost anything you want, followed (3) by the condition that the copyright notice, the permission notes, and the disclaimer "[...] apperar in all copies", and (4) the well known disclaimer[598]. Moreover, as the MIT license, the PGL doesn't talk about the difference of source code and object code. So, you can transfer the MIT analysis[599] to the PGL analogically.

## 6.14 PHP licensed software

The PHP-3.0 license contains a shade more conditions than the MIT license and additionally distinguishes the "redistribution of source code"[600] from the "redistribution in binary form"[601] . Nevertheless, the PHP license focusses only on the redistribution or - as we are going to say in the OSLiC - *the 2others use cases*. Thus, the PHP specific finder can be simplified:

[598] cf. *Open Source Initiative*: PostgreSQL License, 2013, wp.
[599] → OSLiC, p. 175
[600] cf. *Open Source Initiative*: PHP-3.0, 2013, pp. wp. §1.
[601] cf. id., l.c., pp. wp. §2.




### 6.14.1 PHP-C1: Using the software only for yourself

**means** that you are going to use a received PHP software only for yourself and that you do not hand it over to any 3rd party in any sense.

**covers** OSUC-01, OSUC-03, OSUC-06, and OSUC-09[602]

**requires** no tasks in order to fulfill the conditions of the PHP license with respect to this use case:

- You are allowed to use any kind of PHP software in any sense and in any context without any obligations as long as you do not give the software to 3rd parties.

**prohibits** to endorse or promote any service you establish on the base of this privately used software by the name 'PHP'.

[602] For details → OSLiC, pp. 73 - 80

### 6.14.2 PHP-C2: Passing the unmodified software as source code

**means** that you are going to distribute an unmodified version of the received PHP software to 3rd parties – in the form of source code files or as a source code package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02S, OSUC-05S, OSUC-07S[603]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are retained in your package in the form you have received them.
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.3 PHP-C3: Passing the unmodified software as binary

**means** that you are going to distribute an unmodified version of the PHP received software to 3rd parties – in the form of binary files or as a binary package. In this case it is not discriminating to distribute a program, an application, a server, a snippet, a module, a library, or a plugin as an independent or an embedded unit.

**covers** OSUC-02B, OSUC-05B, OSUC-07B[604]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are **reproduced** by your package in the form you have received them[605]. If you compile the binary file on the base of the source code package and

[603] For details → OSLiC, pp. 73 - 78

[604] For details → OSLiC, pp. 74 - 78

[605] Because you are distributing an unmodified binary, you could assume that the copright screens of the application do already what they have to do

if this compilation does not also generate and integrate the licensing files then create the copyright notice, the PHP conditions, and the PHP disclaimer according to the form of the source code package and insert these files into your distribution manually[606].

- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.4 PHP-C4: Passing a modified program as source code

**means** that you are going to distribute a modified version of the received PHP program, application, or server (proapse) to 3rd parties – in the form of source code files or as a source code package.

**covers** OSUC-04S[607]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are retained in your package in the form you have received them.
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the program is licensed under the PHP license. Because you are already modifying the program you can also add such a hint if the presented original copyright notice lacks such a statement. If such a notice is missed in the copyright screen, consider, whether it is possible, to let it *reproduce* the complete PHP license including the

606) For implementing the handover of files correctly → OSLiC, p. 83
607) For details → OSLiC, pp. 75

copyright notice, the PHP conditions, and the PHP disclaimer – as it is required for binary distributions[608]

- [**voluntary:**] Mark your modifications in the source code.

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.5 PHP-C5: Passing a modified program as binary

**means** that you are going to distribute a modified version of the received PHP program, application, or server (proapse) to 3rd parties – in the form of binary files or as a binary package.

**covers** OSUC-04B[609]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.
- [**voluntary:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are **reproduced** by your package. If such a notice is missed in the copyright screen, modify the screen so that it *reproduces* the complete PHP license including the copyright notice, the PHP conditions, and the PHP disclaimer.
- [**voluntary:**] Mark your modifications in the source code, even if you do not want to distribute the code.

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.6 PHP-C6: Passing a modified library as independent source code

**means** that you are going to distribute a modified version of the received PHP code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or as a source code package, but without embedding it into another larger software unit.

608) Following distributors of compiled versions will appreciate your prepatory work.
609) For details → OSLiC, pp. 75

**covers** OSUC-08S[610]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are retained in your package in the form you have received them.
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**voluntary:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.
- [**voluntary:**] Mark your modifications in the source code.

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.7 PHP-C7: Passing a modified library as independent binary

**means** that you are going to distribute a modified version of the received PHP code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of binary files or as a binary package but without embedding it into another larger software unit.

**covers** OSUC-08B[611]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.
- [**voluntary:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are **reproduced** by your package – as far as this can be done by the library itself.
- [**voluntary:**] Mark your modifications in the source code, even if you do want to distribute the code.

610) For details → OSLiC, pp. 79
611) For details → OSLiC, pp. 79

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.8 PHP-C8: Passing a modified library as embedded source code

**means** that you are going to distribute a modified version of the received PHP code snippet, module, library, or plugin (snimoli) to 3rd parties – in the form of source code files or an integrated source code package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10S[612]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are retained in your package in the form you have received them.
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.
- [**voluntary:**] It is a good practice of the open source community, to let the copyright notice which is shown by the running program also state that the program uses a copmponent licensed under the PHP license. So, let the copyright screen of the overarching program *reproduce* the complete PHP license including the copyright notice, the PHP conditions, and the PHP disclaimer – as it is required for binary distributions[613]
- [**voluntary:**] Mark your modifications in the source code.
- [**voluntary:**] Arrange your source code distribution so that the licensing elements – esp. the PHP license text, the specific copyright notice of the original author(s), and the PHP disclaimer – clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the embedded components like libraries, modules, snippets, or plugins in specific directory which contains also all additional licensing elements.

[612] For details → OSLiC, pp. 81

[613] Following distributors of compiled versions will appreciate your prepatory work.

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.9 PHP-C9: Passing a modified library as embedded binary

**means** that you are going to distribute a modified version of the received PHP code snippet, module, library, or plugin to 3rd parties – in the form of binary files or as a binary package together with another larger software unit which contains this code snippet, module, library, or plugin as an embedded component.

**covers** OSUC-10B[614]

**requires** the following tasks in order to fulfill the license conditions:

- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain a line of acknowledgment in the form "This product includes PHP, freely available from ⟨http://www.php.net/⟩".
- [**mandatory:**] Let the documentation of your distribution and/or your additional material also contain the original copyright notice, the PHP conditions, and the PHP disclaimer.
- [**voluntary:**] Ensure that the complete PHP license – esp. the copyright notice, the PHP conditions, and the PHP disclaimer – are **reproduced** by your package, esp. by the copyright screen of your overaching program which uses the library.
- [**voluntary:**] Mark your modifications in the source code, even if you do not you want to distribute the code.
- [**voluntary:**] Arrange your binary distribution so that the licensing elements – esp. the PHP license text, the specific copyright notice of the original author(s), and the PHP disclaimer – clearly refer only to the embedded library and do not disturb the licensing of your own overarching work. It's a good tradition to keep the libraries, modules, snippet, or plugins in specific directories which contain also all licensing elements.

**prohibits** to endorse or promote your product by mentioning PHP, esp. not by make the string 'PHP' part of its name.

### 6.14.10 Discussions and Explanations

First of all, it might surprise some readers that the OSLiC also describes the open source use cases which concern the distribution of binary files although it deals

[614] For details → OSLiC, pp. 81

with the PHP license. PHP is a script language. Thus, delivering the source code seems to be a must. But one has to consider that the PHP license could also be applied to works which are based on other languages constituted on the compiler paradigm. Or there might be used PHP compiler.

It might also surprise some readers that in case of the binary distribution of modifications the condition to repoduce the php license in the documentation is a *must*, while its reproduction in a copyright screen of the program is a *should*. This is directly evoked by the binary-conditon of the php license which expressly requires that "Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution"[615]. But of course, to implement the *must* and the *should* is the best.

[615] cf. *Open Source Initiative*: PHP-3.0, 2013, pp. wp. §2.

# 7 Open Source Licenses and Their Legal Environments [tbd]

*In this chapter we analyze why to know a license alone is not enough. At the end you will know that open source licenses are embedded into the legal environment of a state. And you will know in which sense the German legal environment predetermines your readings of open source licenses.*

# 8 Conclusion

*This chapter shortly describes what the OSLiC is, how it should be used, and how it can be read. It shall be written as top-down explanation.*

# 9 Appendices

## 9.1 Some Additional Remarks on the OSLiC Quotation Style

We have already characterized the general tone of our footnotes[616]. Let us now briefly explain a little peculiarity of our bibliography:

Modern times have also changed the humanities. Formerly a book or an article must be printed for being ripe to be quoted. Our statements relied on static, readily prepared works. Nowadays even university libraries sometimes offer those books and articles as PDF files which are printed in the original. As a scholar, now you must rely on the equality of the printed version and the PDF file - at least with respect to the page numbers and the appearance. You can not verify the equivalence - at least to a certain degree.

Moreover: in case of such 'e-books' and 'e-articles' the libraries often do not offer the pdf files themselves but links to the download pages of the publisher. Formerly as a scholar you could trust that your readers would be able to retrieve the quoted work if they want to verify your citations. It's one task of our libraries to hold available our scientific sources. But now they do not buy any longer the books, but the right to download files over the university net. In this case these PDF files are not stored on the serves of the university library. By using the link provided by the publisher each student or each reader downloads his own file - case by case. Therefore - as a scholar - you now have to trust that the publisher, who provides the link, will not change that pdf file that you have cited.

But it gets even worse: While it might be that publishers modify their work secretly (even it is not very likely that they do it), it's a definite feature of the web that its pages are frequently changed. Hence we must ask ourselves: Can we seriously argue on the basis of statements and documents which might disappear? Can we quote such possibly volatile sources? The problem is: we must do it, especially if we write about an internet topic - and even if we want to write a really reliable compendium.

So, what can we do? First, we must confide in our readers, that they either will retrieve our sources or - if they can not find them - that they believe that we really have found and read what we have written and quoted. Second, we store

616) → p. 13

all these e-wares[617] we read[618]. And thirdly we should lay open to our readers the different levels of reliableness of our sources. Therefore we use the following markers in our bibliographic data[619]:

- Print / Copy:- The source is printed and we saw either the printed work really or we get an official copy by our library. Hence you should also be able to get the work in a library, at least in those we used (UB Frankfurt or ULB Darmstadt).
- BibWeb/[PDF/...] :- The source might be printed, but we read only the electronic version (PDF or other type of format), offered by and over the net of our university libraries (UB Frankfurt or ULB Darmstadt).
- FreeWeb/[PDF/...] :- We read the electronic version offered by the free web. In this case we add the url[620] and the date when we downloaded / saw the text.

## 9.2 Some Widespread Open Source Myths

From the viewpoint of an internet student we have to consider that the web offers a mass of rumors concerning the nature of open source software (Licenses). Here are some of the myths[621] we met:

**BE CAREFUL: THIS SECTION MUST THOROUGHLY BE REVIEWED AND REWRITTEN. IT'S ONLY AN OUTLINE!!! Do not quote part of it. It must be verified**

617) Take this little word as (new) generalization of 'e-book', 'e-article', 'e-paper' and so on.

618) But because of the copyright we ourselves are naturally not allowed to offer a download link for them or to send a copy of it to those who want to verify our quotes.

619) And another hint: Nowadays sometimes even scientific libraries don't offer exact 'e-copies' of the original. In some cases one can only get html-versions of articles which formerly were printed as part of journals. In these case the scholar has to use sources which lost their original page-numbers. The same can happen to articles of proceedings etc. which are now only offered as autonomous pdf files with an internal paging. If we quote such kind of articles we try to specify the number of the quoted article in the original row of articles, added - if possible - by an internal page number. But naturally we also try to follow the bibliographic data delivered by that organization which distributes these kind of copies.

620) Please note: Long urls often destroy the pleasing appearance of a text because it's difficult to wrap the lines acceptably. Hence we wished to make it easier for LaTeX to do this job. Therefor we sometimes split the urls and inserted blanks. So you have to erase all blanks if you want to verify our urls.

621) At least one time even a scientific legally discussing book is talking about the "myth around open source licenses" - although only as part of the title: cf *Guibault, Lucie* a. *Ot van Daalen*: Unravelling the Myth around Open Source Licenses. An Anaysis from A Dutch and European Law Perspective; The Hague: T. M. C. Asser Press, 2006 (= IT & Law, [Vol./No.] 8), ISBN 978–90–6704–214–7, pp. 1ff, especially 209ff.

**open source tries to improve the world ethically** :- No, there's a clear ban to exclude persons, groups, purposes. Thus, there is no chance to exclude anyone from using open source software because he is an ethical or moralic malefactor.

**Changed open source software must be re-published** :- No, in a double sense! There are OS licenses which allow the proprietarization of the modified code. And even the LGPL and the GPL, which clearly try to prevent the proprietarization, do not require generally that a modified code must be (re-)published. Only if you give your modfied (L)GPL licensed application as binary to anyboday, then you have to handover the modified code too.

**Modified open source software must be given back to the whole community** :- No. Again, there are OS licenses which allow the proprietarization of the modified code. And even the LGPL and the GPL - which clearly require, that you also publish the modified code, if you give the modified binary to anybody – do not require that you distribute your modification around the world. LGPL and GPL clearly say that you have to hand over the code to those persons which you give the binary. And if you only give your improvement only one person or a group of person, then you must handover your code only to that persons or only to all members of that group.

**Published open source software is open for ever** :- No, if this myth says that also all future versions will have to be distributed under an open source license. The copyright holder ever holds the copyright. They can change the licence of next release of its software – but only for the following release, not for the current or for former versions. Those releases, which already have been distribuited under an open source license, indeed remain open.

**Software can either be open source software or proprietary software** :- No. The copyright holders themselves can additionally distribute the code under other conditions when ever they want to do it. That's not a question of the licence, but of the copyright.

**The opposite of open source software is commercial Software** :- No. First, you are also allowed to use the open source software in any commercial purpose. There's only one point which is excluded in OSS: you are not allowed to ask for a licence fee if you distribute 'open source software'. Second, there are many other forms like freeware, public domain software or anything else which is neither open source software nor Commercial Software. It's pointless to take the question of money as a criterion for distinguish open source software and its opposite. Moreover: Proprietary Software as opposite of open source software should be defined ex negativo: all kind of software, which does not fit the OSD is proprietary.

**open source software prohibits to earn money** :- No, you are allowed to invent

each business model you want. There's only one exception: you are not allowed to ask for a licence fee if you distribute open source software. This limitation is based on the open source definition which clearly states that a license – which wants to become an open source license – "shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources" and that the license under this circumstances "[...] shall not require a royalty or other fee for such sale"[622]. If you combine this constraint with the requirements that an open source license "[...] must not restrict anyone from making use of the program [...]"[623] and that it "[...] must allow distribution in source code as well as compiled form [...]"[624], you can generally conclude that none of the open source licenses may require a fee for using and/or distributing the program. But being paid for the service to install the programm, to collect and compile a customer specific version, and/or to monitor the environment is of course not excluded by this condition.

Historically this mistake might be evoked by Debian: The GNU project missed its kernel while the Linux kernel was already distributed as part of collections which also include GNU software. Then, in 1983? Ian Murdock was supported by RMS and its FSF to build a really free distribution (Debian) containg GNU software and the Linux kernel. But Ian Murdock states also, that Debian does not want to earn money.

**Modifications of open source software must be marked** :- No. This is not a defining postulation of the OSD. The OSD allows licenses to require the mark of modifications. But it does not require from all licenses to require the mark modifications for being an open source license.

**Modifications of open source software must be marked by your personal data** :- No, it is only required to mark modifications so that a reader could distinguish the modifications from the original code. It's required for saving the integrity of the original author. And therefore it is not required as a constitutiv criterion by the OSD. It might be that a license additionally requires your name. But that is not feature of open source software in general. And at least the licenses discussed by us do not require to insert your name.

**The open source Definition determines the conditions to use open source software** :- No. The *Open Source Definition* determines which licenses are open source licenses, nothing more. The OSD is a set of necessary conditions to be an open source license. It determines the freedom and the respon-

[622] cf. *Open Source Initiative*: The Open Source Definition, 2012, Ω§1.
[623] cf. id., l.c., Ω§6.
[624] cf. id., l.c., Ω§2.

sibilities of a user as a set of more or less abstract rules. But it does not constitute a set of sufficient tasks which a user has to perform for fulfilling any open source license. Open source licenses may differ by instantiating the OSD criteria. So, if you want to know what you have to do to fulfill a license, you have to go back to the real license of that software you are using.

*This section outlines reflections by which we initially focused ourselves on the question why we need an OSLiC and how its content and form should be derivated from these needs.*

### 9.2.1 Why

Do we need another book about open source? Do *you* need another book about open source software? Let us address this question from the viewpoint of what we already know, what we instinctively believe and what we may have heard. For example you may presume one or more of the following statements are correct. Or you may even have experienced similar perceptions from your peers or managers. Or you have been told they describe 'open source':

- The *Open Source Definition* offers rules to use open source software.
- Modified open source software must be published.
- Modified open source software must be given back to the community.
- All generations of open source software will remain open for ever.
- Software can either be open source software or proprietary software.
- The opposite of open source software is commercial software.
- open source software prohibits to earn money.
- Modifications of open source software must be marked explicitly.
- Modifiers of open source software must identify themselves.
- When distributing an open source binary it's enough point to a download page to obtain the source code.
- The aim of open source software is to improve the world ethically.
- open source software is viral and infectious.

Do these conceptions sound familiar to you? Unfortunately, whatever we might believe or wish for, these concepts are incorrect. Naturally we will discuss this issue later on. For the moment let us assume they are indeed incorrect[625].

[625] For those who want directly verify our argumentation, we have generated a condensed summary of the arguments and citations. You can find this summary in our appendices.

So, again: Do *we* need another book about open source software? *We*, that is – in this case and at least initially – the large German company *Deutsche Telekom AG*. Arguing from the perspective of a large company requires not only identifying the common misconceptions, but catering for the unique needs of a large Enterprise. And indeed the very size of the company brings its own problems.

Large companies use more open source software in more varied contexts than small companies. There is an important question that every company should ask: *'Are we sure that we respect all those requirements of open source software we have to respect?'*. But large companies cannot answer this question as easily as small companies: the large number of diverse open source deployments in different contexts mean that case by case governance, a model that may work in small concerns, is far from appropriate for our needs. This leads to wasting both time and money. Further, the chances of success are small: training at least one employee in each software team as an open source software License expert is unrealistic in terms of cost-efficiency and reliability.

Nevertheless even large companies want to and try to fulfill the rules of open source software thoroughly – especially *Deutsche Telekom AG*. When this company realized that the question *Are we sure that we respect all those rules of open source software correctly which we have to respect* could be problematic, it directly asked some of its employees known as open source enthusiasts to establish a service and a process for answering this question.

So, it is no surprise that we, the initial authors of this *Open Source License Compendium*, were asked by our employer *Deutsche Telekom AG*. Naturally we were proud to work on an open source topic officially. But while we were doing our job we had to ask ourselves if *we* perhaps needed another book on open source. Our answer was *Yes, we do!* Let us shortly explain, why:

First, we already knew that there exists supporting software. These meta-programs take the code of any other application and try to list those open source components being 'covered' by that application[626]. But we had also already realised that this supporting software did not always match the way we thought the problem should be solved. Second, we recognized fairly quickly that we need a reliable guide. We personally were asked to give the *ok* for projects of our company. We could not answer such requests on the base of *'Oh yes, I read this in the* Heise-Ticker *a few days ago'* – even if the *Heise-Ticker* had described the situation completely correctly. We ourselves had to be more reliable than this[627].

[626] As general examples let us mention Palamida (`http://www.palamida.com/`) and BlackDuck (`http://www.blackducksoftware.com/`).

[627] But of course, we have to do ourselves the honor of conceding that we – like many many other German open source enthusiasts – love using the *Heise-Ticker* as main IT information source. Unfortunately, its reputation is stil not high enough that its news can directly be cited.

Naturally we already knew a great deal about open source software. Even so, our knowledge was not as systematic as necessary. We looked for an open source compendium which adequately described what a project or product development team had to do to fulfill the criteria of its open source licenses. We wanted to use that compendium to the basis of our recommendations.

We were very thorough but we did not find what we were looking for. Our 'little' bibliography attest our seriousness. What we found was a lot of information releated to individual issues spread over many sources. We did not find answers to our question even in the specific literature. Let us describe three little steps to increase the understanding of the issue:

Without open source licenses there is no open source movement. Nevertheless in dealing with open source licenses, this is sometimes neglected. Take the *Apache Web Server* as an example: No doubt, it is one of the most important pieces of open source software[628] with a specific license[629]. Moreover: the success of the open source movement in the commercial world depends directly on the decision of IBM to replace its corresponding own component in the *IBM WebSphere Application Server* with the free *Apache Web Server*[630]. Meanwhile many companies use the *Apache Web Server* to act as a web provider. Currently the *Apache http server* – as it has to be named correctly – is used more than twice as much as all the other http server software together[631]. Hence many business models depend on the Apache License. Another aspect is that even the famous *Apache Cookbook*, which explains the installation, the configuration, and the maintainance of

[628] To prove that the *Apache* is really a piece of open source software one must execute a set of steps: First, you have to note, that *Apache* is something like a meta project, covered by the *Apache Software Foundation*, also known as *ASF* (cf. `http://www.apache.org/`, wp). Thus, you can not directly jump into the *Apache License*. First of all you have to visit the project site (cf. `http://httpd.apache.org/`, wp) even if at the end its license link leads you back to the general *Apache License sub site* (cf. `http://www.apache.org/licenses/`, wp) which announces, that "all software produced by The Apache Software Foundation or any of its projects or subjects is licensed according to the terms of the documents listed below". Only now you can use the offered link for switching to the *Apache License*, Version 2.0, if you want to check your rights and duties. But that is difficult. There does not exist any simple list what you have to do for fulfilling the license. Even the faq (cf. `http://httpd.apache.org/docs/2.2/faq/`, wp) – meanwhile being moved to a wiki – only says that the server "[...] comes with an unrestrictive license" and that you are allowed to put the code on a CD (cf. `http://wiki.apache.org/httpd/FAQ`, wp). Hence, from the viewpoint of the ASF the license itself shall answer all questions. [Reference download for all urls: 2011-08-31]

[629] cf. *Apache Software Foundation*: Apache License, 2.0, wp.

[630] cf. *Moody*: Die Software-Rebellen, 2001, pp. 287ff.

[631] cf. *Netcraft*: August 2011 Web Server Survey; 2011 ⟨URL: `http://news.netcraft.com/archives/2011/08/05/august-2011-web-server-survey-3.html`⟩ – reference download: 2011-08-31, wp.

an Apache Web Server in details[632], does not mention anything about the license which allows for installation, configuration and maintenance. Neither the index lists the word 'license'[633], nor the chapters 'Installation'[634] or the chapter 'Miscellaneous'[635] mentions the license question in a serious way. There's only one short hint as to the advantage of open source software, i.e. that everybody is allowed to install it[636]. Can you be sure that you are allowed to do what you are doing on the base of such a phrase?

Naturally, the *Apache Cookbook* is not a book for lawyers, it is a book for administrators and developers. They do not want to get bogged down by legalities, they want to set up an Apache Web Server as fast as possible and get down to work. Indeed, the Apache Cookbook offers a good support. But not only as a company you have to ask yourself whether you are really allowed to do what you are doing. Can you find the answer in the *Apache Cookbook*? No. Can you find it in the license itself? Yes, but it is difficult[637]. So again: Can you find your answer in another book, which is *Amazon's* current top recommendation for the search term *'apache server'*[638]? Not really: Sascha Kersken's Apache 2.2 Handbook offers a license chapter, but it is only two pages long[639]. Moreover, the rights and duties are condensed into just 5 bullet points which taken together do not explain when the software and the license have to be handed over to a customer and when you are allowed to hide your improvements[640].

This brings us to the question of what prevents us from using something like a *'general license cookbook'* which explains all the necessary details and which offers quick access to the relevant points:

Of course we also browsed the internet. At least for German speaking people there is an excellent site concerning the topic *open source licenses.* offered by *iffross*, which, loosely translated, means an *Institute for Legal Aspects of the Free and open source software*[641], founded in 2000 as a private institute to track the

632) cf. *Coar, Ken* a. *Rich Bowen*: Apache Kochbuch; deutsche Übersetzung v. Jochen Wiedmann; Beijing [...]: O'Reilly, 2004, ISBN 3–89721–371–0, et passim.
633) cf. id., l.c., pp. 245ff, esp. p. 250.
634) cf. id., l.c., pp. 1ff.
635) cf. id., l.c., pp. 219ff.
636) cf. id., l.c., pp. 1: " ...einer der Vorzüge von open source software besteht darin, dass jedermann die Erlaubnis zur Erzeugung eines eigenen Installationskits hat ".
637) And do we really want our developers and maintainers to read the original licenses? Do we really want them to discover that they also have to check the licenses of the used modules?
638) Tested on `http://www.amazon.de/` at 2011-08-31.
639) cf. *Kersken, Sasche*: Apache 2.2. Das umfassende Handbuch; 3rd, refreshed a. expanded edition; Bonn: Galileo Press, 2009, ISBN 978–8362–1325–7, pp. 111f.
640) cf. id., l.c., p. 112.
641) originally: "Institut für Rechtsfragen der Freien und open source software". Main entry point for its site is the URL `http://www.ifross.org/`.

phenomenon 'free software' from the viewpoint of (German) lawyers[642]. Besides many other aspects this site offers a very well and thoroughly elaborated FAQ[643] and a large list of open source licenses and other related licenses: moreover, evidently it is classifying the open source licenses in those 'without copyleft-effect' (BSD), in those with 'strict copyleft-effect' (GPL)) and in those with 'restricted copyleft-effect' (LGPL)[644].

However, even this excellent site does not fulfill our needs. It does not offer those context specific to-do lists which companies, developers or project managers can use to ensure their open source software is used in a regular manner.

We therefore evaluated that standard book which is listed in the most legal bibliographies[645]: the book of Jaeger and Metzger which concerns – loosely translated – *the judicial framework requirement for open source software*[646]. Even the most earliest edition of this book already had a clear structure in its chapter 'copyright': For each license mentioned (or at least for each license cluster) it offered a subchapter for the rights and a subchapter for the duties[647] of the software user[648]. Many other important aspects of the topic *open source* are discussed, too[649].

But we needed more than this. Despite the quality of the book we were certain that we could not hand over this book to our programmers with the recommendation *check your touched licenses and follow the instructions of the relevant subchapters. . . .* This book did not contain simply checkable to-do lists, neither in the first edition[650] and in the second edition[651] nor in the recently published third edition[652]. So, how can a company or a developer or a project manager be

642) cf. *ifross*: Ziele, Aufgaben, Geschichte; 2011 ⟨URL: `http://www.ifross.org/node/16`⟩ – reference download: 2011-09-05, wp.

643) cf. *ifross*: FAQ; 2011 ⟨URL: `http://www.ifross.org/faq-haeufig-gestellte-fragen`⟩ – reference download: 2011-09-05, wp.

644) cf. *ifross*: ifross Lizenz-Center, 2011, wp.

645) at least in that German judicial literature dealing with open source

646) cf. *Jaeger, Till* a. *Axel Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 1st edition. München: Verlag C.H. Beck, 2002, ISBN 3406484026, pp. V – It can not be any surprise that both authors, Mr. Jaeger and Mr. Metzger are members of ifross (cf. `http://www.ifross.org/personen/`, wp).

647) cf. id., l.c., pp. 30ff.

648) For getting a good survey of the structure and the line of thought see the contents cf. id., l.c., pp. VIIIf.

649) pars pro toto: have a look at the chapter concerning the liability: cf. id., l.c., pp. 137ff.

650) cf. id., l.c., pp. VIff.

651) cf. *Jaeger, Till* a. *Axel Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 2nd edition. München: Verlag C.H. Beck, 2006, ISBN 3406538037, pp. VIIff.

652) cf. *Jaeger a. Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software, 2011, pp. VIIIff. Naturally we use this latest edition for adopting or discussing systematical aspects.

sure of fulfilling the requirements of the open source licenses sufficiently if he/she does not have a verified list telling him *'do this, and in case of that, do that, and finally do also this'*? Why should he himself implicitly become an open source licenses expert who has to extract the necessary steps out of the literature?

While we were searching for an existing open source compendium, we found an article with the title 'Compendium for the Publication of open source software'[653]. It aims to be a 'pragmatic guidebook' and an 'assistance' for 'publishing software under the conditions of an open source license'[654]. Moreover, at the end of this article, its authors formulate ambitiously that their 'guide' should be carried out, section by section – for getting a legally water tight process of publishing open source software[655].

The authors of this article describe something close to what we were looking for. Indeed, the article lists important aspects which have to be taken in consideration if you want to deal open source software correctly: It announces that no obligation exists to publish code either if you embed GPL code into your proprietary code or if you modify the GPL code. It is only if you hand over your binary to other persons that you have to distribute the code too, but only to them and not to the general public[656]. Additionally the articles explains exactly that software – at least in Germany – can only be acknowledged as open source software by transferring the rights to use – the *'Nutzungsrechte'* – to other people, while the copyright itself – the *'Urheberpersönlichkeitsrecht'* – is not transferable and belongs to the author[657]. Moreover, besides other aspects the articles briefly and deeply discusses the problem of the No-Warranty-Clauses which are not valid in Germany and which will therefore automatically be replaced by the liability rules for a donation[658]. And last but not least this article actually summarizes the idea of Copyleft and the differences between LGPL and GPL[659].

However some gaps remain. The article does not analyze in which cases a University or a company perhaps *must* publish its developments based on open source software. It does not discern between different licenses and conditions. It also does not discuss what Universities or companies, which (re-)use and/or distribute

653) approximately translated

654) cf. *Bretschneider, Ulrich*, *Rainer Glaschick*, a. *Gernot Gräfe*: Ratgeber für die Veröffentlichung von Open-Source-Software durch eine Hochschule; In *Asche* et al.: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 166f (originally: ein "pragmatischer Ratgeber" zur "Veröffentlichung einer Software unter den Rahmenbedingungen einer Open-Source-Lizenz").

655) cf. id., l.c., pp. 186 (originally: ein "Ratgeber", der es erlaubt " (...) die zu berücksichtigende Aspekte (strukturiert abzuarbeiten) (...) " und einen "rechtlich nicht angreifbaren Veröffentlichungsprozess" zu ermöglichen).

656) cf. id., l.c., pp. 170 and 181.

657) cf. id., l.c., p. 173.

658) cf. id., l.c., p. 177.

659) cf. id., l.c., p. 181.

open source software (internally), must do to fulfill the touched open source licenses. And finally this article does not offer the step by step list as promised.

We did, however, feel supported by this article, in two ways. First, it was a well written summary of some main problems. Second, it stated the necessity to have a compendium for being able to establish a legally 'water-tight' process of publishing open source software[660]. We seemed to be justified in our assumptions. But the open source compendium we were looking for had to be more practical, more processable, more distinguishing and more elaborated.

So again: Did we need a new book about open source software? We had looked for a reliable integrated open source compendium. But we found separate pieces of information and – as we know today – some rumors. Our answer was clear: naturally we did not need a new general book about open source. But what was lacking was a description of what responsible developers, project managers or product developers require to fulfill open source licenses. We needed an *Open Source License Compendium*.

At the best such an *Open Source License Compendium* would contain a set of simply to process *'For-Fulfilling-The-License-To-Do-Lists'*. Additionally it should offer an intuitively user-friendly search option for these lists. In any case, it should share developers and project managers the effort of having to become open source license experts. For the other users, it should also clearly explain why one has to do this and not that. Hence a reliable *Open Source License Compendium* should not only list what one has to do, but should offer both, thoroughly verified reliable details and clearly condensed guidance.

Although we did not find such an open source compendium we were familiar with the spirit of the open source community. Hence we followed one of its most simple rules: *'what you miss you must develop on your own'*. Some principles should help us to achieve our targets:

**To-do lists as the core, discussions around them** : Our work should be split into two parts. As its core we wanted to offer a set of to-do-Lists. Each of these lists should be relevant to one specific open source license and should be clustered by the open source specific use cases. Around this all those aspects of open source software which influence the interpretation of the licenses and the rules core should be precisely characterized. Nevertheless, the users should be able to skip details and go directly to the section they require.

**Quotations with thoroughly specified sources** : Even if our users should not be obliged to read every part of the compendium they should not be required to believe us. We wanted to be revisable. Because our sources and

[660] cf. *Bretschneider, Glaschick, a. Gräfe*: Ratgeber für die Veröffentlichung von Open-Source-Software durch eine Hochschule, 2008a, p. 186.

our conclusions should be easily verifiable, we decided to use the academic citations and list bibliographic data extensively on the basis that our task should be to collect information, not to invent new 'facts'.

**Not the internet alone, also books and articles** : We wanted to go back to the originals even if the internet was full of more or less modified copies. We wished to get reliable facts and descriptions. Therefore we decided to evaluate not only the internet but also scientific sources – for example – offered by university libraries.

**Not clearing out the forest land, but cutting out a swathe** : Even if we had to deal with licenses and their legal aspects we did not want to get lost in detailed discussions. It should not be our task to find out whether a specific kind of handling would still be legal or already forbidden. We did not want to fight against the licenses. We did not want to stretch their ambit or to test their boundary. We wished to accept open source licenses as they are: rules written from developers for developers. And even if some parts of these licenses would not be valid with respect to a legal system[661], we wanted to take them as our guideline – at least while they do not violate more general laws[662]. We simply wanted to *find one proven way* to cross the maybe slightly unsure forest of open source licenses. Even if indeed some clauses of the licenses finally were not enforceable against us we wanted to respect them 'voluntarily'. We wanted to deliver a set of rules which support users and remove the possibility of becoming involved in license disputes with open source developers or the Free Software Foundation.

**Take the text seriously** : On the other side we wanted to take our license texts as they were. If they lacked anything[663], we would interpret the open issues

661) And indeed for example for the GPL one can argue in this way: Even if you take the GPL as a contract of the type 'donation' respectively "Schenkung", it is presented in the form of AGBs respectively "Allgemeine Geschäftsbedingungen" and must therefore follow the general AGB rules.'Regrettably' in Germany these general AGB rules do not allow to exclude each type of warranty. If we follow Oberhem, §11 and §12 of the GPL must be invalid in Germany because of these general AGB rules. Moreover, for Oberhem even §5 – the important clause of the GPL by which you can only get the right to use and to distribute GPL software if you respect the rules of the GPL – seems also to be invalid respectively "unwirksam". But the good message is that the GPL as whole is not invalid even if it contains invalid clauses. *Oberhem, Carolina*: Vertrags- und Haftungsfragen beim Vertrieb von Open Source Software; Dissertation; Hamburg: Verlag Dr. Kovač, 2008 (= Recht der Neuen Medien, [Vol./No.] 50), ISBN 978–3–8300–4075–0, pp. 128, 133ff, 150ff, esp. 146, 159.

662) what they clearly do not do!

663) The systematical underdetermination of licenses is a problem being also known in the open source respectively Free Software movement. Following the biography of RMS his main judicial counselor Moglen has stated, that "there is uncertainty in every legal process (...) " and that it seemed to be silly to try "(...) to take out all the bugs (...)". Nevertheless – so Moglen resp. Williams – the goal of Richard Stallman was "the complete opposite": He

in the spirit of the open source idea. But where the text was clear and definite we wanted to take its propositions as a definite decision – even if that meaning stood against well known open source 'facts'.

**Trust the swarm** : We did not want to use our own research alone as a basis. We knew that the swarm is ever stronger than a set of some randomly selected experts. Therefore we decided to publish our text as a still unfinished work, starting with an early release 0.2. And then we wanted to invite the community to complete the compendium together with us. We would elaborate our open source compendium as a set of LaTeX- and BibTeX files which could be developed and managed in GIT or any other version control system. And finally we would publish our text under a Creative Commons Attribution-Share Alike German 3.0 license, to allow other people to correct us, to help us or even to take our results for their own purposes.

And so we did. Here is the result. Feel free to use it – according to our licensing.

### 9.2.2 What

Now we can briefly explain how one should be able to use the compendium:

**The Same Idea, Different Licenses** :- Here you will find background information to help you interpret open source licenses in the sense of the *Free Software movement*[664], the *open source software movement*[665], or the GNU-Project[666]. We discuss different ways to cluster open source licenses. Finally we present our own taxonomy based on the labels 'protecting the developer',

tried "(...) to remove uncertainty which is inherently impossible". But – and that's the nub of this analysis – Moglen had to follow Stallmann because of RMS character. And he had to summarize their work so, that "(...) the resulting elegance (of the GPL; KR.), the resulting simplicity (of the GPL; KR.) in design almost achieves what it has to achieve". Hence we are asked to take the license texts themselves seriously. cf. *Williams*: Free as in Freedom. Richard Stallman's Crusade for Free Software, 2002, pp. 177f.

664) At least at this place you are perhaps expecting that we use the logograms FLOSS, F/OSS, F/LOSS, or whatever. As you will read later on the word *Free* is ambiguous and has strained the use of the concept *Free Software*. Later on we will also talk about the invention of the concept *open source* designed as a 'replacement' and acting as a 'splitter'. The mentioned logograms are introduced to re-establish or – at least – to underline the common history and the common center of 'both' movements, whereby the word *Libre* shall resolve the ambiguity of the word *Free*. For a first survey cf. *Wikipedia (en)*: Free and open source software; n.l., 2011 ⟨URL: `http://en.wikipedia.org/wiki/Free_and_open_source_software`⟩ – reference download: 2011-09-08, wp.

665) For another brief and informative introduction cf. *Fogel*: Producing Open Source Software, 2006, pp. 231ff esp. p. 232f.

666) We ourselves will stay with the concept *open source* because the OSD specifies the scope of our analysis. But we do it with a deep obeisance to Stallmann and the FSF – even if we know that this will not protect us from the thunderbolt of RMS.

'protecting the licensed code' and 'protecting the on-top-developments'. If you are familiar with the methods of grouping different open source licenses and particular if you know that you can not authorize your doings on the base of descriptions of such license groups, then it is enough, in order to understand our line of thought, to briefly note our taxonomy and its wording.

**The Problem of Derivated Works** :- This chapter is important. In the spirit of software developers we try to explain which kinds of programming evoke a derivated work and which not. Our to-do lists will refer to this analysis.

**The Problem of Combining Different Licenses** :- You should not ignore this chapter. We will explain why and how combining software of different licenses is not as dangerous as it is often told. The results of this chapter influence the structure of our to-do lists.

**open source software and Money** :- Here we will shortly discuss ways in which money is no problem. If you already know that it is only prohibited to require payment for the act of licensing a piece of open source software to second or third parties and if you already know that this is only forbidden by some licenses, and not by all, than you can postpone the reading of this chapter.

**The Problem of Implicitly Freeing Patents** :- Here we will illuminate some aspects of software patents and how the are handled by some open source licenses. You should know what licenses implicitly do with your patents. But it is not our intention to write a software patent compendium.

**Open Source Use Cases as Principle of Classification** :- This is an important chapter. We explain our categories 'Use as it is', 'Modify the Code', 'With Redistribution', 'Without Redistribution', 'Isolated Initial Development', 'On-Top-Development': we develop and discuss our taxonomy with respect to the side effects of 'combining different licenses' and 'generating derivated works'. This taxonomy will determine the following chapters.

**open source licenses: Find Your Specific To-do Lists** :- This is a kind of summary which joins the relevant aspects and elaborates the 'finder for your to-do lists'. This is the chapter which you probably will reuse frequently, even if you do not want to read any of our explanations.

**open source license Fulfillment: Classified To-do Lists** :- This chapter offers all classified to-do lists. The structure of its subchapters will match the structure of our finder and the structure of our taxonomy.

**open source licenses and Their Legal Environments** :- Here we discuss why using open source software in a regular manner is not only a question of the licenses themselves but of the kind of the surrounding legal system.

**Appendices: Some Widespread Open Source Myths** :- Here we make good on our promise to explain why all the propositions mentioned at the beginning of this chapter are wrong. You might read this chapter as a special introduction or a reminder epilogue whenever you want to do.

# Periodicals, Shortcuts, and Abbreviations

| | |
|---|---|
| AGPL | GNU Affero General Public License |
| ApL | Apache License |
| BISE | Business & Information Systems Engineering [ISSN: 1867-0202] |
| BSD | Berkeley Software Distrobution (License) |
| [n.abbr.] | Berkeley Technology Law Journal |
| BWV | Berliner Wissenschafts-Verlag GmbH |
| [n.abbr.] | Cultural Anthropology [ISSN: 1548-1360] |
| CiHB | Computers in Human Behavior [ISSN: 0747-5632] |
| CotACM | Communications of the ACM [ISSN: 0001-0782] |
| CR | Computer und Recht. Zeitschrift für die Praxis des Rechts der Informationstechnologien |
| CRi | Computer Law Review international [ISSN: 1610-7608] |
| [n.abbr.] | Computers & Education [ISSN: 0360-1315] |
| [n.abbr.] | Cutter IT Journal [ISSN: 1048-5600] |
| DDT | Drug Discovery Today [ISSN: 1359-6446] |
| DSS | Decision Support Systems [ISSN: 0167-9236] |
| [n.abbr.] | Ethics and Information Technology [ISSN: 1388-1957] |
| E.C.L.R. | European Competition Law Review |
| EER | European Economic Review [ISSN: 0014-2921] |
| EPL | Eclipse Public License |
| et seqq. | and the following ones |
| EUPL | European Union Public License |
| GPL | GNU General Public License |
| [n.abbr.] | Information & Management [ISSN: 0378-7206] |
| ibid. | ibidem = latin for 'at the same place' |
| ICC | Industrial and Corporate Change [ISSN: 0960-6491] |
| id. | idem = latin for 'the same', be it a man, woman or a group... |
| IEaP | Information Economics and Policy [ISSN: 0167-6245] |
| [n.abbr.] | IEEE Software [ISSN: 0740-7459] |
| ifross | Institut für Rechtsfragen der Freien und Open Source Software |
| [n.abbr.] | International Information and Library Review [ISSN: 1057-2317] |
| [n.abbr.] | International Journal of Medical Informatics [ISSN: 1386-5056] |
| [n.abbr.] | interactions[ISSN: 1072-5520] |
| ISJ | Information Systems Journal [ISSN: 1365-2575] |
| ITRB | Der IT-Rechtsberater [ISSN: 1617-1527] |
| JAIS | Journal of the Association for Information Systems [ISSN: 1536-9323] |
| JCSC | Journal of Computing Sciences in [Small] Colleges [ISSN: 1937-4771] |
| JISE | Journal of Information Science and Engineering [ISSN: 1016-2364] |
| JLEO | Journal of Law, Economics, & Organization [ISSN: 1465-7341] |
| JMIR | Journal of Medical Information Research [ISSN: 1438-8871] |
| [n.abbr.] | Journal of Academic Librarianship [ISSN: 0099-1333] |

| | |
|---|---|
| [n.abbr.] | Journal of Comparative Economics [ISSN: 0147-5967] |
| [n.abbr.] | Journal of Systems and Software [ISSN: 0164-1212] |
| JSIS | Journal of Strategic Information Systems [ISSN: 0963-8687] |
| l.c. | loco citato = latin for 'in the place cited' |
| LGPL | GNU Lesser General Public License |
| LJ | Linux Journal [ISSN: 1075-3583] |
| MIT | Massachusetts Institute of Technology (License) |
| MPL | Mozilla Public License |
| Ms-PL | Microsoft Public License |
| n.abbr. | no abbreviation (known) |
| [n.abbr.] | netWorker [ISSN: 1091-3556] |
| n.y. | year not stated / no year |
| n.l. | location not stated / no location |
| np. | no page numbering |
| n.st. | not stated |
| [n.abbr.] | Organization Science [ISSN: 1047-7039] |
| PgL | Postgres License |
| PHP | PHP (License) |
| [n.abbr.] | Queue [ISSN: 1542-7730] |
| [n.abbr.] | R&D Management [ISSN: 1467-9310] |
| RP | Research Policy [ISSN: 0048-7333] |
| SIGCSE Bulletin | SIGCSE Bulletin [ISSN: 0097-8418] |
| SIGCAS | ACM SIGCAS Computers and Society [ISSN: 0095-2737] |
| SIGMIS Database | ACM SIGMIS - The Data Base for Advances in Information Systems [ISSN: 0095-0033] |
| SIGSOFT SEN | SIGSOFT Software Engineering Notes [ISSN: 0163-5948] |
| [n.abbr.] | Stanford Law Review [ISSN: 00389765] |
| [n.abbr.] | Software Qualilty Journal [ISSN: 0963-9314] |
| STHV | Science, Technology & Human Values [ISSN: 0162-2439] |
| ToIT | Transaction on Internet Technology [ISSN: 1533-5399] |
| ToSEM | Transactions on Software Engineering Methodology [ISSN: 1049-331X] |
| Ubiquity | Ubiquity - The ACM IT Magazine and Forum [ISSN: 1530-2180] |
| UB | 'Universitätsbibliothek' = library of university X |
| ULB | 'Universitäts- & Landesbibliothek' = library of university and state X |
| [n.abbr.] | University of Chicago Law Review |
| [n.abbr.] | University of Illinois Law Review |
| [n.abbr.] | University of Pittsburgh Law Review |
| wp. | webpage / webdocument without any internal (page)numbering |
| ZGE / IPJ | Zeitschrift für geistiges Eigentum [ISSN: 1867-237x] |

# Bibliography


Perspectives on free and open source software; o.O.?: ???, 2005
*[Präsenz FaM Bibliothek Recht und Wirtschaft 05/ST 230 F318] [bibliographic data have be verified]*

Proceedings of the Linux Symposium; Ottawa, 2006
*[bibliographic data have be verified]*

*Ågerfalk, Pär et al., editors*: Open Source Software: New Horizons; 6th International IFIP WG 2.13 Conference on Open Source Systems, OSS 2010; (= IFIP Advances in Information and Communication Technology, [Vol./No.] 319) Berlin, Heidelberg u. New York: Springer, 2010, BibWeb/PDF, ISBN 978–3–642–13243–8
*eval copies*

*Ahtiainen, Aleksi*, *Sami Surakka*, a. *Mikko Rahikainen*: Plaggie: GNU-licensed Source Code Plagiarism Detection Engine for Java Exercises; in: Proceedings of the 6th Baltic Sea Conference on Computing Education Research; New York, NY, USA: ACM, 2006 (= Baltic Sea '06) ⟨URL: `http://doi.acm.org/10.1145/1315803.1315831`⟩ – reference download: 2011-12-29, BibWeb/PDF, pp. 141–142
*[to be evaluated]*

*Airlie, D. M.*: Open Source Graphic Drivers. They Don't Kill the Kittens; In Proceedings of the Linux Symposium, 2006, p. 19.26
*[bibliographic data have be verified]*

*Alexi, O.* a. *J. Henkel*: Promoting the Penguin: Who Is Advocating Open Source Software in Commercial Settings? München, 2007 ⟨URL: `http://ssrn.com/abstract/=988363`⟩
*next action, aus Simone Käs*

*Alexy, Oliver*: Free Revealing. How Firms Can Profit From Being Open; Wiesbaden: Gabler, 2009 (= Gabler Edition Wissenschaft), Print and BibWeb/PDF, ISBN 978–3–8349–1475–0
*[tbd: evaluate copy]*

*Allman, Eric*: A Conversation with Chris DiBona; in: Queue, 1 July (2003), pp. 10–19 ⟨URL: `http://doi.acm.org/10.1145/945074.945130`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Allman, Eric* a. *Marshall Kirk McKusick*: From the Editors: Open Source Revisited; in: Queue, 2 May (2004), pp. 8–9 ⟨URL: `http://doi.acm.org/10.1145/1005062.1005072`⟩
*[to be evaluated]*

*Alspaugh, Thomas A.*, *Hazeline U. Asuncion*, a. *Walt Scacchi*: Analyzing software licenses in open architecture software systems; In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009, pp. 54–57 ⟨URL: `http://dx.doi.org/10.1109/FLOSS.2009.5071361`⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Alspaugh, Thomas A.*, *Walt Scacchi*, a. *Hazeline U. Asuncion*: Software Licenses in Context: The Challenge of Heterogeneously-Licensed Systems; in: JAIS, 11 (2010), No. 11/12, pp. 730–755, BibWeb/PDF
*[to be evaluated]*

*Amega-Selorm, Charles* a. *Johanna Awotwi*: Free and Open Source Software (FOSS): It's Significance or Otherwise to the E-Governance Process in Ghana; in: Proceedings of the 4th International Conference on Theory and Practice of Electronic Governance; New York, NY, USA: ACM, 2010 (= ICEGOV '10) ⟨URL: http://doi.acm.org/10.1145/1930321.1930342⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0058–2, pp. 91–95
*[to be evaluated]*

*anonymous*: Mircosoft und Open Source: nichts als Ärger; 2000 ⟨URL: http://www.heise.de/newsticker/meldungen/8314⟩
*[bibliographic data have be verified]*

*anonymous*: Microsoft und die GPL: Freiheit, die ich meine ... 2002 ⟨URL: http://www.heise.de/newsticker/meldung/26355⟩
*[bibliographic data have be verified]*

*anonymous*: Mircosoft Shared Source Initiative Overview; 2004 ⟨URL: http://www.microsoft.com/resources/sharedsource/Intiative/Intiative.mspx⟩
*next action - aus buchtala [bibliographic data have be verified]*

*Anonymous*: The Triumph of the Commons; in: The Economist (2005)
*next action, aus buchtala [bibliographic data have be verified]*

*Anvaari, Mohsen* a. *Slinger Jansen*: Evaluating Architectural Openness in Mobile Software Platforms; in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume; New York, NY, USA: ACM, 2010 (= ECSA '10) ⟨URL: http://doi.acm.org/10.1145/1842752.1842775⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0179–4, pp. 85–92
*[to be evaluated]*

*Apache Software Foundation*: Apache License, Version 2.0; 2004, FreeWeb/Html ⟨URL: http://www.apache.org/licenses/LICENSE-2.0⟩ – reference download: 2011-08-31
*The orignal text of the apache license. Specifies the 'Does' and the 'Don'ts'.*

*Apache Software Foundation*: Licenses; 2013 [n.y ], FreeWeb/Html ⟨URL: http://www.apache.org/licenses/⟩ – reference download: 2013-02-25
*Tells the history of the Apache Licenses and marks the releases 1.0 and 1.1 as 'historical' - only release 2.0 has been classified as an open source license by the OSI.*

*Arlt, Brinkel*, a. *Volkmann; Spindler, Gerald, editor*: 'BSD' - und 'Mozilla'-artige Lizenzen; In *Spindler*: Rechtsfragen bei Open Source Software, 2004, pp. 317–372, Print
*This chapter of the book describes the thoughts of the BSD and the Mozilla licenses. The Mozilla licenses are taken as Copyleft licenses which differ from the GPL.*

*Arnö, Kaj*: Dual Licensing - A Business Model from the Second Generation of Open-Source Companies; In *Wynants a. Cornelius*: How Open is the Future?, 2005, pp. 479–486
*next action, aus buchtala [bibliographic data have be verified]*

*Asay, Matt*: A Funny Thing Happened on the Way to the Market: Linux, the General Public License, and a New Model for Software Innovation; Stanford CA, 2002, Web/Pdf ⟨URL: http://www.linuxdevices.com/files/misc/asay-paper.pdf⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Asche, Michael et al., editors*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen; (= POWeR / Patent Offensive Westfalen Ruhr, [Vol./No.] 3) Münster, New York, München [... etc.]: Waxmann, 2008, Print, ISBN 978–3–8309–1845–5
*This collection focuses on patents and Open Source Software as a challenge for innovative universities. It offers articles analyzing different aspects of this topic.*

*Ascher, David*: Is Open Source Right for You? in: Queue, 2 May (2004), pp. 32–38 ⟨URL: http://doi.acm.org/10.1145/1005062.1005065⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Asiri, Sami*: Open Source Software; in: SIGCAS, 33 March (2003), p. 2 ⟨URL: http://doi.acm.org/10.1145/966498.966501⟩ – reference download: 2011-12-28, BIbWeb/HTML
*[to be evaluated]*

*Asundi, Jai*: The Need for Effort Estimation Models for Open Source Software Projects; In Proceedings of the Fifth Workshop on Open Source Software Engineering, 2005, pp. 6:1–6:3 ⟨URL: http://doi.acm.org/10.1145/1082983.1083260⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Axel Metzger, Till Jaeger und*: Die neue Version 3 der GNU Genereal Public License; in: GRUR, o.A. (2008), No. 2, pp. 130–137, Copy
*fernleihe*

*Ayala, Claudia et al.*: Challenges of the Open Source Component Marketplace in the Industry; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 213–224, BibWeb/PDF
*[tbd: evaluate copy]*

*Azzi, R. Michael*: CPR: How Jacobsen V. Katzer Resuscitated the Open Source Movement; in: University of Illinois Law Review, (2010), No. 4, pp. 1271–1302, BibWeb/PDF
*[to be evaluated]*

*Babcock, Charles*: Big Test For Open Source GPL; in: Informationweek, 17 December (2006), p. np., Copy
*The very short article refers to the famous case 'Jacobsen v. Katzer' and states, that it only covered the Artitistic License. For the GPL and under the same interpretation the Free Software Foundation would get the right 'to stop Cisco from using the code' - a big test.*

*Bach, Paula M.* a. *John M. Carroll*: Characterizing the Dynamics of Open User Experience Design: The Cases of Firefox and OpenOffice.org; in: JAIS, 11 (2010), No. 12, pp. 902–925, BibWeb/PDF
*[to be evaluated]*

*Backu, Frieder*: Open Source Software und Interoperabilität; in: ITRB (IT-Rechtsberater), (2003), p. 180
*aus Koeglin2007a [bibliographic data have be verified]*

*Baerwolff, Matthias*, *Robert A. Gehring*, a. *Bernd Lutterbeck, editors*: Open Source Jahrbuch 2005. Zwischen Softwareentwicklung und Gesellschaftsmodell; Berlin: Lehmanns Media, 2005 ⟨URL: http://www.opensourcejahrbuch.de/download/jb2005/OpenSourceJahrbuch2005_online.pdf⟩ – reference download: 2011-10-17, Print & FreeWeb/PDF, ISBN 3–86541–059–6
*Second volume of the famous German written row 'Open Source Annual'. Offers articles on heterogeneous aspects of Open Source Software. [tbd: evaluate copy]*

*Bain, Malcolm et al.*: Legal Aspects of the Information Society; Oberta de Catalunya: Free Technology Academy, 2010 ⟨URL: http://www.ftacademy.org/materials/fsm/6#1⟩ – reference download: 2012-101-20, FreeWeb/PDF
*[to be evaluated]*

*Baird, Stacy Avery*: The Heterogeneous World of Proprietary and Open-Source Software; in: Proceedings of the 2nd international conference on Theory and practice of electronic governance; New York, NY, USA: ACM, 2008 (= ICEGOV '08) ⟨URL: http://doi.acm.org/10.1145/1509096.1509143⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–60558–386–0, pp. 232–238

*[to be evaluated]*

*Bake, Pio*: Open Source Software, Competition and Potential Entry; Working Paper; 2003, Web/Pdf ⟨URL: `http://www.berlecon.de/tw/osscompetition.pdf`⟩
*next action, aus buchtala*

*Bakker, Arno*, *Maarten Van Steen*, a. *Andrew S. Tanenbaum*: A Wide-Area Distribution Network for Free Software; in: Transaction on Internet Technology, 6 August (2006), No. 3, pp. 259–281 ⟨URL: `http://doi.acm.org/10.1145/1151087.1151089`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Baldi, Stefan*, *Hauke Heier*, a. *Anett Mehler-Bicher*: Open Courseware and Open Source Software; in: Communications of the ACM, 46 September (2003), No. 9, pp. 105–107 ⟨URL: `http://doi.acm.org/10.1145/903893.903922`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Barcellini, Flore*, *Françoise Détienne*, a. *Jean Marie Burkhardt*: Cross-Participants: Fostering Design-Use Mediation in an Open Source Software Community; in: Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore! New York, NY, USA: ACM, 2007 (= ECCE '07) ⟨URL: `http://doi.acm.org/10.1145/1362550.1362564`⟩, ISBN 978–1–84799–849–1, pp. 57–64
*[to be evaluated]*

*Barcellini, Flore et al.*: Thematic Coherence and Quotation Practices in OSS Design-Oriented Online Discussions; in: Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work; New York, NY, USA: ACM, 2005 (= GROUP '05) ⟨URL: `http://doi.acm.org/10.1145/1099203.1099237`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 1–59593–223–2, pp. 177–186
*[to be evaluated]*

*Bayersdorfer, Mitch*: Managing a Project With Open Source Components; in: interactions, 14 November/December (2007), pp. 33–34 ⟨URL: `http://doi.acm.org/10.1145/1300655.1300677`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Baytiyeh, Hoda* a. *Jay Pfaffman*: Open source software: A community of altruists; in: Computers in Human Behavior, 26 (2010), p. 1345–1354, BibWeb/PDF
*[to be evaluated]*

*Beard, Ashley* a. *Hyunju Kim*: A Survey On Open Source Software Licenses; in: JCSC, 22 (2007), No. 4, pp. 205–211 ⟨URL: `http://dl.acm.org/citation.cfm?id=1229637.1229673`⟩
*[to be evaluated]*

*Behlendorf, Brian*: Open Source as a Business Strategy; In *DiBona, Ockman, a. Stone*: Open Sources, 1999, pp. 149–170
*[bibliographic data have be verified]*

*Behlendorf, Brian*: How Open Source Can Still Save the World; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, p. 2, BibWeb/PDF
*Keynote. Affirms that the style of cooperation being offered by Open Source Software supports also the management of general crises.*

*Beigbeder, Michel*, *Wray Buntine*, a. *Wai Gen Yee*: Open Source Search and Research; in: Proceedings of the 2006 International Workshop on Research Issues in Digital Libraries; New York, NY, USA: ACM, 2007 (= IWRIDL '06) ⟨URL: `http://doi.acm.org/10.1145/1364742.1364748`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–59593–608–4, pp. 5:1–5:7
*[to be evaluated]*

*Berglund, Erik* a. *Michael Priestley*: Open-Source Documentation: In Search of User-Driven, Just-in-Time Writing; in: Proceedings of the 19th annual international conference on Computer documentation; New York, NY, USA: ACM, 2001 (= SIGDOC '01) ⟨URL: http://doi.acm.org/10.1145/501516.501543⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–295–6, pp. 132–141
*[to be evaluated]*

*Bergquist, Magnus* a. *Jan Ljungberg*: The power of gifts: organizing social relationships in open source communities; in: Information Systems Journal, 11 (2001), pp. 305–320
*next action, aus buchtala [bibliographic data have be verified]*

*Berlecon[-]Research*: Basics of Open Source Software Markets and Business Models; FLOSS Final Report - Part 3; 2002, Web/Pdf ⟨URL: http://www.berlecon.de/studien/downloads/200207FLOSS_Basics.pdf⟩
*next action , aus buchtala [bibliographic data have be verified]*

*Berlecon[-]Research*: Firms Open Source Activities: Motivations and Policy Implications; FLOSS Final Report - Part 2; 2002, Web/Pdf ⟨URL: http://www.berlecon.de/studien/downloads/200207FLOSS_Activities.pdf⟩
*next action , aus buchtala [bibliographic data have be verified]*

*Berry, David M.*: Copy, Rip, Burn; The Politics of Copyleft and Open Source; London: Pluto Press, 2008, Print, ISBN 978–0–7453–2414–2
*This sociological book contains at least some remarks on the open source history - focusing on the internal 'battles' ('from software to open source?') and its discources ('the contestation of code').*

*Bessen, James*: What Good is Free Software? In *Hahn*: Government Policy toward Open Source Software, 2002, pp. 12–34
*next action, aus buchtala [bibliographic data have be verified]*

*Bessen, James*: Holdup and licensing of cumulative innovations with private information; in: Economics Letters, 82 (2004), No. 3, pp. 321–326
*next action, aus buchtala [bibliographic data have be verified]*

*Bessen, James* a. *Robert M. Hunt*: An empirical look at software patents; in: Federal Reserve Bank of Philadelphia, Working Paper 03-17 (2004)
*next action, aus buchtala [bibliographic data have be verified]*

*Bezroukov, Nikoplai*: BSD vs. GPL: Part 2: The Dynamic Properties of BSD and GPL Licenses in the Context of the Program Life Cycle; 2003 ⟨URL: http://www.softpanorama.org/Copyright/License-classification/social_dynamics_of_BSD_and_GPL.shtml⟩
*next ection, aus buchtala [bibliographic data have be verified]*

*Bhattacharya, Jaijit* a. *Sourabh Suman*: Analysis of popular open source licenses and their applicability to e-governance; in: Proceedings of the 1st international conference on Theory and practice of electronic governance; New York, NY, USA: ACM, 2007 (= ICEGOV '07) ⟨URL: http://doi.acm.org/10.1145/1328057.1328110⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–59593–822–0, pp. 254–257
*[to be evaluated]*

*Bianco, Vieri del et al.*: The QualiSPo approach to OSS product quality evaluation; [General Chairs: Justin Erenkrantz and Hyrum K. Wright]; In Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2010, pp. 23–28 ⟨URL: http://doi.acm.org/10.1145/1833272.1833277⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Bieman, James*: Editorial: Free/open source software, silver bullets, and mythical months; in: Software Quality Journal, 14 (2006), pp. 289–290 ⟨URL: http://dx.doi.org/10.1007/s11219-006-0036-3⟩ – reference download: 2012-202-03, BibWeb/PDF

*[to be evaluated]*

*Bird, Christian*, *Alex Gourley*, a. *Prem Devanbu*: Detecting Patch Submission and Acceptance in OSS Projects; in: Proceedings of the Fourth International Workshop on Mining Software Repositories; Washington, DC, USA: IEEE Computer Society, 2007 (= MSR '07) ⟨URL: `http://dx.doi.org/10.1109/MSR.2007.6`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 0–7695–2950–X, pp. 26:1–26:4
*[to be evaluated]*

*Bird, Christian et al.*: Open Borders? Immigration in Open Source Projects; in: Proceedings of the Fourth International Workshop on Mining Software Repositories; Washington, DC, USA: IEEE Computer Society, 2007 (= MSR '07) ⟨URL: `http://dx.doi.org/10.1109/MSR.2007.23`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 0–7695–2950–X, pp. 6:1–6:8
*[to be evaluated]*

*Bird, Christian et al.*: Latent Social Structure in Open Source Projects; in: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering; New York, NY, USA: ACM, 2008 (= SIGSOFT '08/FSE-16) ⟨URL: `http://doi.acm.org/10.1145/1453101.1453107`⟩ – reference download: 2012-01-02, BibWeb/PDF, ISBN 978–1–59593–995–1, pp. 24–35
*[to be evaluated]*

*Bitzer, Jürgen*, *Wolfram Schrettl*, a. *Philipp J.H. Schröder*: Intrinsic motivation in open source software development; in: Journal of Comparative Economics, 35 (2007), p. 160–169, BibWeb/PDF
*[to be evaluated]*

*Björgvinsson, Tryggvi* a. *Helgi Thorbergsson*: Software Development for Governmental Use Utilizing Free and Open Source Software; in: Proceedings of the 1st International Conference on Theory and Practice of Electronic Governance; New York, NY, USA: ACM, 2007 (= ICEGOV '07) ⟨URL: `http://doi.acm.org/10.1145/1328057.1328087`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–59593–822–0, pp. 133–140
*[to be evaluated]*

*Böckler, Lina*: Mit Freier Software gegen den Wettbewerb; in: *Katharina Vera Boesche, editor:* Variationen im Recht: Beiträge zum Arbeits-, Immaterialgüter-, Infrastruktur-, Lauterkeits-, Unternehmens-, Wettbewerbs- und Zivilrecht; Festbeigabe für Fanz Jürgen Säcker zum 65. Geburstag; Berlin: BWV, 2006, ISBN 3–8305–1234–1, pp. 69 – 76, BibWeb/Copy
*[eval copy]*

*Boehm, Barry*: A View of 20th and 21st Century Software Engineering; in: Proceedings of the 28th International Conference on Software Engineering; New York, NY, USA: ACM, 2006 (= ICSE '06) ⟨URL: `http://doi.acm.org/10.1145/1134285.1134288`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–59593–375–1, pp. 12–29
*[to be evaluated]*

*Boldrin, Michele* a. *David K. Levine*: The Case Against Intellectual Property; in: American Economic Review, 92 (2002), No. 2, pp. 209–212
*next action, aus buchtala [bibliographic data have be verified]*

*Boldrito, Remo Suppi* a. *Josep Jorba Esteve*: GNU/Linux Advanced Administration; coordinated by Josep Jorba Esteve; Barcelona: Free Technology Academy, 2007 ⟨URL: `http://www.ftacademy.org/`⟩ – reference download: 2012-01-20, FreeWeb/PDF
*This book offers some short conceptual summaries before it deeply explains how to administrate a GNU/Linux system. It's also important because it's published under the GNU Documentation License and/or under the Creative Commons License. [tbd. eval electronic file]*

*Boldyreff, Cornelia et al., editors*: Open Source Ecosystems: Diverse Communities Interaction; 5th IFIP WG 2.13 International Conference on Open Source Systems, OSS 2009 Skövde,

Sweden, June 3-6, 2009; Berlin, Heidelberg and New York: Springer, 2009, BibWeb/PDF, ISBN 978–3–642–02031–5

*Proceedings. Relevant articles are itemized explicitly.*

*Bolzern, Mark*: A New Project or a GNU Project? in: Linux Journal, 13 May (1995), p. 7:1 ⟨URL: `http://dl.acm.org/citation.cfm?id=324822.324829`⟩ – reference download: 2011-12-29, BibWeb/HTML

*eval copy*

*Bonaccorsi, A.*, *M. Merito ans L. Piscitello*, a. *C. Rossi*: The 'Open Innovation' Paradigm. How Firms Do Business out of Open Source Software; Copenhagen, 2006, Paper preseneted at the DRUID Summer Conference

*next action, aus Simone Käs [bibliographic data have be verified]*

*Bonaccorsi, A.* a. *C. Rossi*: Contributing to the Common Pool Resources in Open Source Software. A Comparison between Individuals and Firms. Pisa, 2003, Sant' Anna School of Advanced Studies; Working Paper

*next action, aus Simone Käs [bibliographic data have be verified]*

*Bonaccorsi, A.* a. *C. Rossi*: Why Open Source Software Can Succeed; in: Research Policy, 32 (2003), No. 7, pp. 1243–1258

*next action, aus Simone Käs [bibliographic data have be verified]*

*Bonaccorsi, A.* a. *C. Rossi*: Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement. From Community to Business; Pisa, 2004, Sant' Anna School of Advanced Studies; Working Paper ⟨URL: `http://opensource.mit.org/`⟩

*next action, aus Simone Käs [bibliographic data have be verified]*

*Bonaccorsi, Andrea et al.*: Business Firms' Engagement in Community Projects. Empirical Evidence and Further Developments of the Research; in: Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development; Washington, DC, USA: IEEE Computer Society, 2007 (= FLOSS '07) ⟨URL: `http://dx.doi.org/10.1109/FLOSS.2007.3`⟩, ISBN 0–7695–2961–5, pp. 13–

*[bibliographic data have be verified]*

*Booth, David R.*: Peer Production and Software. What Mozilla Has To Teach Government; Cambridge (Massachusetts) and London (England): MIT Press, 2010 (= The John D. and Catherine T. MacArthur Foundation Reports on Digital Media and Learning), Print, ISBN 978–0–262–51461–3

*The book describes some aspects of the collaboration in the Mozilla project and its objective Firefox.*

*Bresson, Jean*, *Carlos Agon*, a. *Gérard Assayag*: OpenMusic; Visual Programming Environment for Music Composition, Analysis and Research; in: Proceedings of the 19th ACM International Conference on Multimedia; New York, NY, USA: ACM, 2011 (= MM '11) ⟨URL: `http://doi.acm.org/10.1145/2072298.2072434`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0616–4, pp. 743–746

*[to be evaluated]*

*Bretschneider, Ulrich*, *Rainer Glaschick*, a. *Gernot Gräfe*: Ratgeber für die Veröffentlichung von Open-Source-Software durch eine Hochschule; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 167–188, Print

*This article summarizes basic aspects of an Open Source publication succesfully. It addresses the (German) difference between copyright ('Urheberecht') and transferring the right to use ('Nutzungsrecht'). It mentions the problem of liability. It discusses the choice of an Open Source License with respect to the indented purpose and many things more. Nevertheless the article can't be taken as the sought-after 'Open Source Compendium': it doesn't analyze*

*in which cases a University perhaps must publish its' developments or what it must do for fulfilling the licenses of internally (re-)used and distributed Open Source Software.*

*Brodie, Mark et al.*: Support services: persuading employees and customers to do what is in the community's best interest; in: Proceedings of the 2nd international conference on Persuasive technology; Berlin, Heidelberg: Springer-Verlag, 2007 (= PERSUASIVE'07) ⟨URL: `http://dl.acm.org/citation.cfm?id=1780402.1780424`⟩, ISBN 3–540–77005–4, 978–3–540–77005–3, pp. 121–124
*[bibliographic data have be verified]*

*Brown, Peter*: EOF: The Free Software Foundation at 20; in: Linux Journal, 137 September (2005), p. 15:1 ⟨URL: `http://dl.acm.org/citation.cfm?id=1084783.1084798`⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Brügge, Bernd et al.*: Open-Source Software. Eine ökonomische und technische Analyse; Berlin and Heidelberg: Springer, 2004, Print, ISBN 3–540–20366–4
*This book wants to explain central factors of OSS: it lists examples, clusters OS licenses, describes the development process and analyzes the OSS work of companies. But nevertheless it does not describe in a real sense what companies have to do for fulfilling the licenses. For the authors these obligations seem to be implicitely clear.*

*Buchtala, Rouven*: Determinanten der Open Source Software-Lizenzwahl. Eine spieltheoretische Analyse; Frankfurt am Main, Berlin, Bern [... etc.]: Peter Lang, 2007 (= Informationsmanagement und strategische Unternehmensführung), [Vol./No.] 12), Print, ISBN 978–3–631–57114–9
*The book tries to detect why specific OS-licenses are chosen. Game theoretic aspects shall help to answer the question. Thereto OS-licenses are classified as 'permissive', 'restrictive' and 'highly restrictive' licenses. This differentiation highlights some necessary constraints for respecting a license. But the names of these categories are generated with respect to a company which wants to be able to (re)use and (re)sell OS code with minimal restrictions. The intention of the OS licensors to defend the licensed freedom for revocations is not covered by these names.*

*Burgess, Guy*: Open Source: The Affero General Public License; in: Magazine of the Society for Computers and Law, 19 (2008), No. 4, pp. 42–43, Copy
*Fernleihe*

*Bygott, David*: David Bygott's Gnu Book. A light-hearted look at the Gnu, or Wildebeest; firstly published 1998; Southerton, Harare: Robert Woollacott, 1992, Print, ISBN 0–7974–1082–1
*Just a little witty comic book playing with the pronounciation of 'GNU' and 'nu' or 'new': it illustrates the 'Revegnue' or the 'Sgnuker'.*

*Bärwolff, M.*: Tight Prior Open Source Equilibrium: The Rise of Open Source as a Source of Economic Welfare; in: First Monday, 11 (2006) ⟨URL: `http://firstmanday.org/issues/issue11_1/barwolff/index.html`⟩
*aus Simone Käs [bibliographic data have be verified]*

*Camp, L. Jean*: DRM: Doesn't Really Mean Digital Copyright Management; in: Proceedings of the 9th ACM Conference on Computer and Communications Security; New York, NY, USA: ACM, 2002 (= CCS '02) ⟨URL: `http://doi.acm.org/10.1145/586110.586122`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–612–9, pp. 78–87
*[to be evaluated]*

*Campbell-Kelly, Martin*: Historical Reflections: Will the Future of Software be Open Source? in: Communications of the ACM, 51 October (2008), No. 10, pp. 21–23 ⟨URL: `http://doi.acm.org/10.1145/1400181.1400189`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Capiluppi, Andrea*, *Andres Baravalle*, a. *Nick W. Heap*: From "Community" to "Commercial" FLOSS - the Case of Moodle; [General Chairs: Justin Erenkrantz and Hyrum K. Wright]; In Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2010, pp. 11–16 ⟨URL: `http://doi.acm.org/10.1145/1833272.1833275`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Capiluppi, Andrea* a. *Thomas Knowles*: Software Engineering in Practice: Design and Architecture of FLOSS Systems; 5th IFIP WG 2.13 International Conference on Open Source Systems, OSS 2009 Skövde, Sweden, June 3-6, 2009; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 34–46, BibWeb/PDF
*[tbd: evaluate copy]*

*Chawner, Brenda*: F/OSS in the Library World: An Exploration; In Proceedings of the Fifth Workshop on Open Source Software Engineering, 2005, pp. 3:1–3:4 ⟨URL: `http://doi.acm.org/10.1145/1082983.1083262`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Cheliotis, Giorgos*: From open source to open content: Organization, licensing and decision processes in open cultural production; in: Decision Support Systems, 47 (2009), No. 3, pp. 229–244 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S0167923609000578`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Chen, Shun-ling*: Free/Open Source Software. Licensing; Shri Pratap Udyog, Sriniwas Puri, New Delhi: Elsevier India, 2006 ⟨URL: `http://www.iosn.net/licensing/foss-licensing-primer/foss-licensing-final.pdf`⟩ – reference download: 2013-02-02, FreeWeb/PDF, ISBN 978–81–312–0422–1
*This e-book shortly delineates 'Free Software' and 'Open Source Software' and then it describes the act of licensing itself more thoroughly.*

*Chen, Zhixiong* a. *Delia Marx*: Experiences with Eclipse IDE in Programming Courses; in: JCSC, 21 December (2005), No. 2, pp. 104–112 ⟨URL: `http://dl.acm.org/citation.cfm?id=1089053.1089068`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Cheung, Gifford et al.*: Designing for Discovery: Opening the Hood for Open-Source End User Tinkering; in: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems; New York, NY, USA: ACM, 2009 (= CHI EA '09) ⟨URL: `http://doi.acm.org/10.1145/1520340.1520660`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–247–4, pp. 4321–4326
*[to be evaluated]*

*Chindalia, Sanjanaa*: Open source software: The future ahead; in: JOURNAL OF INTELLECTUAL PROPERTY RIGHTS, 13 (2008), pp. 218–224
*next action, found in ISI Web of Knowledge [bibliographic data have be verified]*

*Chopra, S.* a. *S. Dexter*: Free software and the political philosophy of the cyborg world; in: SIGCAS, 37 November (2007), No. 2, pp. 41–52 ⟨URL: `http://doi.acm.org/10.1145/1327325.1327328`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Chopra, S.* a. *S. Dexter*: Free software, economic 'realities', and information justice; in: SIGCAS, 39 December (2009), No. 3, pp. 12–26 ⟨URL: `http://doi.acm.org/10.1145/1713066.1713067`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Chopra, Samir* a. *Scott Dexter*: The freedoms of software and its ethical uses; in: Ethics and Information Technology, 11 (2009), pp. 287–297 ⟨URL: `http://dx.doi.org/10.1007/s10676-009-9191-0`⟩, BibWeb/PDF

*[to be evaluated]*

*Christ, Fabian* a. *Stefan Sauer*: OSS - Open-Source-Stacks; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 133–154, Print
*OSS is a favored abbreviation: It doesn't stand only for 'Open Source Software' but for 'Open Source Stacks' or even for 'Operation Support Systems'.*

*Ciffolilli, Andrea*: The Economics of Open Source Hijacking and Declining Quality of Digital Information Resources: A Case for Copyleft; 2004 ⟨URL: `http://opensource.mit.edu/papers/ciffolili.pdf`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Coar, Ken* a. *Rich Bowen*: Apache Kochbuch; deutsche Übersetzung v. Jochen Wiedmann; Beijing [...]: O'Reilly, 2004, Print, ISBN 3–89721–371–0
*Significantly the 'Apache License' which has enabled the wide use of this http server is not discussed in this book. It doesn't ask wether you are allowed to use the software or under which conditions you may use which module, although these answers are necessary to do really what the book describes.*

*Colazo, Jorge* a. *Yulin Fang*: Impact of License Choice on Open Source Software Development Activity; in: Journal of the American Society for Information Science and Technology, 60 (2009), No. 5, pp. 997–1011, BibWeb/PDF
*[tbd: evaluate copy]*

*Coleman, Gabriella*: CODE IS SPEECH: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers; in: Cultural Anthropology, 24 (2009), No. 3, pp. 420–454 ⟨URL: `http://dx.doi.org/10.1111/j.1548-1360.2009.01036.x`⟩ – reference download: 2012-02-03, BibWeb/PDF, ISSN 1548–1360
*[to be evaluated]*

*Comino, Stefano* a. *Fabio M. Manenti*: Dual licensing in open source software markets; in: Information Economics and Policy, 23 (2011), No. 3–4, pp. 234–242 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S016762451100028X`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Comino, Stefano*, *Fabio M. Manenti*, a. *Maria Laura Parisi*: From planning to mature: On the success of open source projects; in: Research Policy, 36 (2007), No. 10, pp. 1575–1586 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S0048733307001709`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Costa-Soria, Cristóbal* a. *Jennifer Pérez*: Teaching Software Architectures and Aspect-Oriented Software Development using Open-Source Projects; in: Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education; New York, NY, USA: ACM, 2009 (= ITiCSE '09) ⟨URL: `http://doi.acm.org/10.1145/1562877.1563027`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–381–5, p. 385
*[to be evaluated]*

*Crowston, Kevin et al.*: Effective Work Practices for Software Engineering: Free/Libre Open Source Software Development; in: Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research; New York, NY, USA: ACM, 2004 (= WISER '04) ⟨URL: `http://doi.acm.org/10.1145/1029997.1030003`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–988–8, pp. 18–26
*[to be evaluated]*

*Crowston, Kevin* a. *James Howison*: The social structure of Free and Open Source software development; in: First Monday 10 (2005), No. 2

*next action, aus buchtala [bibliographic data have be verified]*

*Crowston, Kevin et al.*: Self-organization of teams for free/libre open source software development; in: Information and Software Technology, 49 (2007), No. 6, pp. 564 – 575 ⟨URL: http://www.sciencedirect.com/science/article/pii/S0950584907000080⟩, ¡ce:title¿Qualitative Software Engineering Research¡/ce:title¿, ISSN 0950–5849
*[bibliographic data have be verified]*

*Cuéllar, Luis E.*: Open Source License Alternatives for Software Applications; Is it a solution to stop software piracy?; in: Proceedings of the 43rd Annual Southeast Regional Conference; Volume 2, New York, NY, USA: ACM, 2005 ⟨URL: http://doi.acm.org/10.1145/1167253.1167314⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 1–59593–059–0, pp. 269–274
*[to be evaluated]*

*Currion, Paul*, *Chamindra de Silva*, a. *Bartel Van de Walle*: Open Source Software For Disaster Management; in: Communications of the ACM, 50 March (2007), No. 3, pp. 61–65 ⟨URL: http://doi.acm.org/10.1145/1226736.1226768⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Cusumano, Michael A.*: Reflections on Free and Open Software; in: Communications of the ACM, 47 October (2004), No. 10, pp. 25–27 ⟨URL: http://doi.acm.org/10.1145/1022594.1022615⟩ – reference download: 2012-202-03, BibWeb/PDF
*[to be evaluated]*

*Dahlander, L.*: Appropriation and Appropriability in Open Source Software; in: International Journal of Innovation Management, 9 (2005), No. 3, pp. 259–285
*aus Simone Käs [bibliographic data have be verified]*

*Dahlander, Linus*: Penguin in a new suit: a tale of how de novo entrants emerged to harness free and open source software communities; in: Industrial and Corporate Change, 16 (2007), No. 5, pp. 913–943 ⟨URL: http://icc.oxfordjournals.org/content/16/5/913.abstract⟩ – reference download: 2012-202-03, BibWeb/PDF
*[to be evaluated]*

*Dalle, Jean-Michel et al.*: Advancing Economic Research on the Free and Open Source Software Mode of Production; In *Wynants a. Cornelius*: How Open is the Future?, 2005, pp. 395–426
*next action aus buchtala [bibliographic data have be verified]*

*David, Paul A.* a. *Francesco Rullani*: Dynamics of innovation in an "open source" collaboration environment: lurking, laboring, and launching FLOSS projects on SourceForge; in: Industrial and Corporate Change, 17 (2008), No. 4, pp. 647–710, BibWeb/PDF
*[to be evaluated]*

*Davies, Julius*: Measuring Subversions: Security and Legal Risk in Reused Software Artifacts; in: Proceedings of the 33rd International Conference on Software Engineering; New York, NY, USA: ACM, 2011 (= ICSE '11) ⟨URL: http://doi.acm.org/10.1145/1985793.1986025⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–4503–0445–0, pp. 1149–1151
*[to be evaluated]*

*Davis, Donald* a. *Iffat Jabeen*: Learning in the GNU/Linux Community; in: Proceedings of the 2011 Conference on Information Technology Education; New York, NY, USA: ACM, 2011 (= SIGITE '11) ⟨URL: http://doi.acm.org/10.1145/2047594.2047600⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–1017–8, pp. 21–26
*[to be evaluated]*

*Davis, Mike et al.*: Linux and Open Source in the Academic Enterprise; in: Proceedings of the 28th annual ACM SIGUCCS conference on User services: Building the future; New York,

NY, USA: ACM, 2000 (= SIGUCCS '00) ⟨URL: http://doi.acm.org/10.1145/354908.354923⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 1–58113–229–8, pp. 65–69
*[to be evaluated]*

*de Laat, Paul B.*: Copyright or copyleft? An analysis of property regimes for software development; in: Research Policy, 34 (2005), pp. 1511–1532, BibWeb/PDF
*[tbd: evaluate copy]*

*De Nicolò, Christopher*: Open Source Software - Rechtliche Aspekte nach deutschem und italienischem Recht. Eine rechtsvergleichende Studie. Dissertation; Regensburg: Universität Regensburg, 2010, Print
*Tries to describe the differences between the Italien and the German handling of Open Source Software: With respect to the copyright the OS licenses offer a nearly equal level of proctection. But the type of the contract seems to be differently classified: In Germany Copyleft licenses shall be seen as tradeoff contracts, all other OS licenses as a donation. In Italy contracts of donation are a problem. In return in both countries the licenses are equivalent with respect to the liability.*

*Debian*: The Debian Free Software Guidelines (DFSG); 2013 [n.y.], FreeWeb/HTML ⟨URL: http://www.debian.org/social_contract#guidelines⟩ – reference download: 2013-01-22
*The Debian Free Software Guideline contains nine criteria for being free software, These criteria are embedded into the Debian Social Contract and are also adopted by the Open Source Definition.*

*Deike, Thies*: Open Source Software: IPR-Fragen und Einordnung ins deutsche Rechtssystem; in: CR [Computer und Recht], (2003), pp. 9ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Deitcher, Avi*: The challenges of open source in the enterprise; in: Linux Journal, 195 July (2010), pp. Article No. 3 ⟨URL: http://dl.acm.org/citation.cfm?id=1883478.1883481⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Dempsey, Bert J. et al.*: A quatitative profile of a community of Open Source Linux developers; North Carolina: University of North Carolina at Chapel Hill, School of Information and Librabry Science, 1999 (= (= [University of North Carolina] Technical Report TR 1999-05))
*next action, aus buchtala [bibliographic data have be verified]*

*Dempsey, Bert J. et al.*: Who Is an Open Source Software Developer? in: Communications of the ACM, 45 (2002), No. 2, pp. 67–72 ⟨URL: http://doi.acm.org/10.1145/503124.503125⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Deodhar, Swanand J., K. B. C. Saxena*, a. *Mikko Ruohonen*: Firm-Oriented Success Factors of an Open Source Software (OSS) Product; [General Chairs: Justin Erenkrantz and Hyrum K. Wright]; In Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2010, pp. 1–4 ⟨URL: http://doi.acm.org/10.1145/1833272.1833273⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Determann, Lothar*: Softwarekombinationen unter der GPL; in: GRUR Int. (slg: Gewerblicher Rechtsschutz und Urheberrecht, Internationaler Teil, 2006), (2006), pp. 645 – 653
*aus Koeglin2007a [bibliographic data have be verified]*

*Di Penta, Massimiliano et al.*: An Exploratory Study of the Evolution of Software Licensing; in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering; Volume 1, New York, NY, USA: ACM, 2010 ⟨URL: http://doi.acm.org/10.1145/

1806799.1806824⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–60558–719–6, pp. 145–154
*[to be evaluated]*

*DiBona, Chris*, *Cooper C*, a. *D. Stone, editors*: Open Sources 2.0: The Continuing Evolution; Sebastopol CA: O'Reilly, 2005
*important: many articels die ich nicht einzelnd aufgeschlossen habe [bibliographic data have be verified]*

*DiBona, Chris*, *Sam Ockman*, a. *Mark Stone, editors*: Open Sources. Voices from the Open Source Revolution; Beijing u.a.: O'Reilly, 1999
*Präsenz: FaM, Bibliothek Informatik K.4-89 [bibliographic data have be verified]*

*Diedrich, Oliver*: Die Geschichte von Linux; 2011, FreeWeb/PDF ⟨URL: http://www.heise.de/open/artikel/Die-Geschichte-von-Linux-1329997.html⟩ – reference download: 20110826
*[tbd. evaluate ecopy]*

*Dionisio, John David N. et al.*: An Open Source Software Culture in the Undergraduate Computer Science Curriculum; in: SIGCSE Bulletin, 39 June (2007), No. 2, pp. 70–74 ⟨URL: http://doi.acm.org/10.1145/1272848.1272888⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Djordjevic, Valle et al., editors*: Urheberrecht im Alltag. Kopieren, bearbeiten, selber machen; Bonn: Bundeszentrale für politische Bildung, 2008, Print, ISBN 978–3–89331–812–4
*German written collection of articles which describe aspects of the copy right in the internet. It is not focused on 'Open Source' but on the copy of electronical goods in general. Nevertheless the book mentions the topics '(Open Source) software', 'free culture' and 'creative commons (licences)'.*

*Dobb, Dr.*: It All About The License; in: Informationweek, n.V. (2009), No. 1253, p. 46, Copy *Fernleihe*

*Doernhoefer, Mark*: Surfing the Net for Software Enginerring Notes; in: SIGSOFT Software Engineering Notes, 35 (2010), No. 4, pp. 8–16, BibWeb/PDF
*[eval]*

*Donnelly, Francis P.*: Evaluating open source GIS for libraries; in: Library Hi Tech, 28 (2010), No. 1, pp. 131–151 ⟨URL: http://www.emeraldinsight.com/0737-8831.htm⟩ – reference download: 2012-02-13, BibWeb/PDF
*[to be evaluated]*

*Donorfio, Brian*: The Politics of "Free": Open Source Software in Government; in: JCSC, 19 (2004), No. 5, pp. 279–280 ⟨URL: http://dl.acm.org/citation.cfm?id=1060081.1060117⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Dorman, David*: The Case for Open Source Software in the Library Market; in: Ubiquity, January (2004), p. 4:1 ⟨URL: http://doi.acm.org/10.1145/985600.985601⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Dougherty, William C.* a. *Audrey Schadt*: Linux Is for Everyone; Librarians Included! in: The Journal of Academic Librarianship, 36 (2010), No. 2, pp. 173–175 ⟨URL: http://www.sciencedirect.com/science/article/pii/S0099133310000108⟩ – reference download: 2012-02-09, BibWeb/PDF
*[to be evaluated]*

*Douglas, David*: A bundle of software rights and duties; in: Ethics and Information Technology, 13 (2011), pp. 185–197 ⟨URL: http://dx.doi.org/10.1007/s10676-010-9229-3⟩ – reference download: 2012.02.09, BibWeb/PDF

*[to be evaluated]*

*Drossou, Olga*, *Stefan Krempl*, a. *Andreas Poltermann*: Der Kampf um die Innovationsfreiheit: Der Big Bang des Wissens und seine Sprengkraft. Plädoyer für einen offenen Umgang mit Wissen im Interesse der Innovationskraft von Wirtschaft und Gesellschaft; Editorial; In *Drossou, Krempl, a. Poltmann*: Die wunderbare Wissensvermehrung, 2006, pp. 1–10, Print
*This editorial states that collaboratively supported knowledge must be free if it shall evoke an augmentation of knowledge. Example for the new method is the Open Source collaboration which has already changed the style of cooperating of writers (GNU Free Documentation License, Creative Commons-Initiative). Patents are opposed to that method. But the editorial states that innovation needs a free interplay of forces and an exchange of knowledge as bride as possible.*

*Drossou, Olga*, *Stefan Krempl*, a. *Andreas Poltmann, editors*: Die wunderbare Wissensvermehrung. Wie Open Innovation unsere Welt revolutioniert; (= Telepolis) Hannover: Heise, 2006, Print, ISBN 3–936931–38–0
*A collection of articles which highlight the necessity of a free exchange for innovation emphaticly.*

*Eckl, Julian*: Die politische Ökonomie der Wissenschaftsgesellschaft. Geistige Eigentumsrechte und die Frage des Zugangs zu Ideen; Marburg: Tectum Verlag, 2004, Print, ISBN 3–8288–8735–X
*This book wants to show that for ever the concept of intellectual properties has been discussed controversially and that Open Source therefore challanges the current 'state of the art'-position legitimately. It contains a short but meaningful summary of the Open Source history.*

*Eclipse Foundation*: Eclipse Public License, Version 1.0; 2005 [n.y. of the page itself], FreeWeb/HTML ⟨URL: `http://www.eclipse.org/org/documents/epl-v10.php`⟩ – reference download: 2013-02-20
*The Eclipse Public License 1.0, as it is presented by the Eclipse Foundation.*

*Eclipse Foundation*: CPL to EPL Conversion; 2013 [n.y. of the page itself], FreeWeb/HTML ⟨URL: `http://www.eclipse.org/legal/cpl2epl/`⟩ – reference download: 2013-02-20
*This page collects information concerning the 'transition from the CPL to the EPL' started in 2004 and completed in 2005.*

*Economides, Nicholas* a. *Evangelos Katsamakas*: Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry; in: Management Science, 52 (2006), No. 7, pp. 1057–1071
*next action, aus buchtala: sollte die Frage der Lizenzerfüllung in der Praxis erfüllen, wenn nicht, anzeichen für nicht existenz [bibliographic data have be verified]*

*Elkemann-Reusch, Ilva*: Die erzwungene Gegengabe; in: ZGE / IPJ, 2 (2010), pp. 413–452, Copy
*Fernleihe*

*Elliott, Margaret*, *Mark S. Ackerman*, a. *Walt Scacchi*: Knowledge Work Artifacts: Kernel Cousins for Free/Open Source Software Development; in: Proceedings of the 2007 international ACM conference on Supporting group work; New York, NY, USA: ACM, 2007 (= GROUP '07) ⟨URL: `http://doi.acm.org/10.1145/1316624.1316650`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–59593–845–9, pp. 177–186
*[to be evaluated]*

*Elliott, Margaret S.* a. *Walt Scacchi*: Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration; in: Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work; New York, NY, USA: ACM, 2003 (= GROUP '03) ⟨URL: `http://doi.acm.org/10.1145/958160.958164`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–693–5, pp. 21–30

*[to be evaluated]*

*Ellis, Jason* a. *Jean-Paul Van Belle*: Open Source Software Adoption by South African MSEs: Barriers and Enablers; in: Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers' Association; New York, NY, USA: ACM, 2009 (= SACLA '09) ⟨URL: `http://doi.acm.org/10.1145/1562741.1562746`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–683–0, pp. 41–49

*[to be evaluated]*

*Engelfriet, Arnoud*: Tools of the Trade[:] Choosing an Open Source License; in: IEEE Software, 27 (2010), No. 1, pp. 48–49, Copy

*Fernleihauswertung*

*Epplin, J.*: Using GPL Software in Embedded Applications; ⟨URL: `http://www.linux.devices.com/articles/AT916119242.html`⟩

*aus Simone Käs [bibliographic data have be verified]*

*Ernst, Stefan*: Die Verfügbarkeit des Sourcecodes; in: MultiMedia und Recht, 4 (2001), pp. 208–213

*[aus widmer, next action] [bibliographic data have be verified]*

*Euler, Ellen*: Creative Commons: Mehr Innovation durch die Öffnung des Urheberechts? In *Drossou, Krempl, a. Poltmann*: Die wunderbare Wissensvermehrung, 2006, pp. 147–158, Print

*The article states that the German 'Urheberrecht' was developed to protect the common interest in new ideas: it should solve the free rider problem. Only the author has the right to allow or forbid the distribution of his works. And this right stimulates to create / describe new ideas. But this right is limited by the also true interests of the community. Therefore in Germany private citizens are ever allowed to cite a work or to make private copies of the work. But in our digital world DRM etc. disable this common right and undermine the 'Urheberrecht'. Creative Commons gives the possibility to grant rights back to the author.*

*European Community* a. *European commission Joinup*: European Union Public Licence v. 1.1. 2007, FreeWeb/HTML ⟨URL: `http://joinup.ec.europa.eu/system/files/EN/EUPL%20v.1.1%20-%20Licence.pdf`⟩ – reference download: 2013-02-08

*The English version of the EPL - due to the OSI (http://opensource.org/licenses/EUPL-1.1) also the translations have been certified*

*European Community* a. *European commission Joinup*: New FSF statements on the EUPL are a step in the right direction; 2013 [n.y], FreeWeb/HTML ⟨URL: `https://joinup.ec.europa.eu/community/eupl/news/new-fsf-statements-eupl-are-step-right-direction`⟩ – reference download: 2013-03-05

*The answer of the EU to the FSF statement concerning the limited copyleft effect of the EUPL-1.1*

*Europäische Gemeinschaft* a. *European commission Joinup*: Open-Source-Lizenz für die Europäische Union; 2007, FreeWeb/HTML ⟨URL: `http://joinup.ec.europa.eu/system/files/DE/EUPL%20v.1.1%20-%20Lizenz.pdf`⟩ – reference download: 2013-02-08

*The German version of the EPL - due to the OSI (http://opensource.org/licenses/EUPL-1.1) also the translations have been certified*

*Evans, David S.* a. *Bernard J. Reddy*: Government Preferences for Promoting Open-Source Software: A Solution in Search of a Problem; in: 9 Mich Telecomm. Tech. L. Rev. 313 (2003), pp. 313–394

*next action, aus buchtala [bibliographic data have be verified]*

*Ezeala, Adanna*, *Hyunju Kim*, a. *Loretta A. Moore*: Open Source Software Development: Expectations and Experience from a Small Development Project; in: Proceedings of the 46th Annual Southeast Regional Conference on XX; New York, NY, USA: ACM,

2008 (= ACM-SE 46) ⟨URL: http://doi.acm.org/10.1145/1593105.1593168⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–105–7, pp. 243–246
*[to be evaluated]*

*Fantl, Stephen*: Copyleft or Copyright. Into the new paradigm; in: MacTech Magazine, 16 (2000), No. 10, pp. 98–100, Copy
*fgernleihe*

*Fehr, Ernst* a. *Simon Gächter*: Fairness and Retaliation: The Economics of Reciprocity; in: Journal of Economic Perspectives, 14 (2000), pp. 159–181
*next action, aus buchtala [bibliographic data have be verified]*

*Feig, Michael*: Einführung in GNU; München and Wien: Carl Hanser Verlag, 1996 (= Unix easy), Print, ISBN 3–446–18311–6
*This elder little book concerns the GNU project as a collection of free tools which together shall constitute a free Unix. The GNU license and philosophy are very briefly discussed. But the GNU tools - mostly running in a shell at this time - are described in a deeper sense. Hence in a way this book is something like a compendium for the elder GNU applications.*

*Feller, Joseph*: Meeting challenges and surviving success: the 2nd workshop on open source software engineering; evaluate the complete workshop results; in: Proceedings of the 24th International Conference on Software Engineering; New York, NY, USA: ACM, 2002 (= ICSE '02) ⟨URL: http://doi.acm.org/10.1145/581339.581436⟩, ISBN 1–58113–472–X, pp. 669–670
*[bibliographic data have be verified]*

*Feller, Joseph* a. *Brian Fitzgerald*: A Framework Analysis of the Open Source Software Development Paradigm; in: Proceedings of the twenty first international conference on Information systems; Atlanta, GA, USA: Association for Information Systems, 2000 (= ICIS '00) ⟨URL: http://dl.acm.org/citation.cfm?id=359640.359723⟩, BibWeb/PDF, pp. 58–69
*[to be evaluated]*

*Feller, Joseph et al.*: Collaboration, Conflict and Control: The 4th Workshop on Open Source Software Engineering; evaluate worksup results; in: Proceedings of the 26th International Conference on Software Engineering; Washington, DC, USA: IEEE Computer Society, 2004 (= ICSE '04) ⟨URL: http://dl.acm.org/citation.cfm?id=998675.999508⟩, ISBN 0–7695–2163–0, pp. 764–765
*[bibliographic data have be verified]*

*Feller, Joseph et al.*: Collaboration, conflict and control: report on the 4th workshop on open source software engineering; in: SIGSOFT Softw. Eng. Notes, 30 May (2005), pp. 1–2 ⟨URL: http://doi.acm.org/10.1145/1061874.1061885⟩, ISSN 0163–5948
*evaluate workshop results [bibliographic data have be verified]*

*Feller, Joseph et al.*: Taking stock of the bazaar: the third workshop on open source software engineering; in: SIGSOFT Softw. Eng. Notes, 28 November (2003), pp. 5–5 ⟨URL: http://doi.acm.org/10.1145/966221.966227⟩, ISSN 0163–5948
*evaluatre the workshop results [bibliographic data have be verified]*

*Feller, Joseph*, *Brian Fitzgerald*, a. *André van der Hoek*: 1st workshop on open source software engineering; evaluate workshop results; in: Proceedings of the 23rd International Conference on Software Engineering; Washington, DC, USA: IEEE Computer Society, 2001 (= ICSE '01) ⟨URL: http://dl.acm.org/citation.cfm?id=381473.381660⟩, ISBN 0–7695–1050–7, pp. 780–781
*[bibliographic data have be verified]*

*Feller, Joseph*, *Brian Fitzgerald*, a. *André van der Hoek*: Making sense of the bazaar: 1st workshop on open source software engineering; in: SIGSOFT Softw. Eng. Notes, 26 November (2001), pp. 51–52 ⟨URL: http://doi.acm.org/10.1145/505532.505543⟩, ISSN 0163–5948
*evaluate the complete proceedings [bibliographic data have be verified]*

*Feller, Joseph et al.*: Open source application spaces: the 5th workshop on open source software engineering; evaluatre workshop results; in: Proceedings of the 27th international conference on Software engineering; New York, NY, USA: ACM, 2005 (= ICSE '05) ⟨URL: `http://doi.acm.org/10.1145/1062455.1062619`⟩, ISBN 1–58113–963–2, pp. 694–694
*[bibliographic data have be verified]*

*Feller, Jospeph* a. *Brian Fitzgerald*: Understanding Open Source Software Development; Reading, Mass., London?: Addison-Wesley, 2002
*[in DA bestellen, FaM nicht vorhanden] next action, aus buchtala (important, neuere Version möglich)? [bibliographic data have be verified]*

*Fielding, Roy T.*: Shared leadership in the Apache Project; in: Communications of the ACM, 42 (1999), No. 4, pp. 42–43
*next action, aus buchtala [bibliographic data have be verified]*

*Fielding, Roy T.*: Software Architecture in an Open Source World; in: Proceedings of the 27th International Conference on Software Engineering; New York, NY, USA: ACM, 2005 (= ICSE '05) ⟨URL: `http://doi.acm.org/10.1145/1062455.1062474`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–963–2, pp. 43–43
*[to be evaluated]*

*Fink, Martin*: The business and economics of Linux and open source; Upper Saddle River, N.J.: Prentice Hal, 2003
*next action, aus buchtala [bibliographic data have be verified]*

*Fitzgerald, Brian*: The Transformation of Open Source Software; in: MIS Quarterly, 30 (2006), No. 3, pp. 587–598
*next action, aus buchtala [bibliographic data have be verified]*

*Fogel, Karl*: Producing Open Source Software; How to Run a Successful Free Software Project; Beijing, Cambridge, Köln [...]: O'Reilly, 2006, Print, ISBN 978–0–596–00759–1
*Well known english written standard work. Focuses on the process of developing Open Source Software. Licences are discussed in a small chapter. Describes very clearly the concepts of 'Free Software', 'Open Source Software', 'FLOSS' and so on. Analyzes also the topic 'License Compatibility'.*

*Fosfuri, Andrea*, *Marco S. Giarratana*, a. *Alessandra Luzzi*: The Penguin Has Entered the Building: The Commercialization of Open Source Software Products; in: Organization-Science, 19 March-April (2008), No. 2, p. 292–305, BibWeb/PDF
*[to be evaluated]*

*Fox, Laurie* a. *Shawn Plummer*: Opening the Lines of Communications with Open Source Software; in: Proceedings of the 34th Annual ACM SIGUCCS Fall Conference; New York, NY, USA: ACM, 2006 (= SIGUCCS '06) ⟨URL: `http://doi.acm.org/10.1145/1181216.1181242`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–59593–438–3, pp. 114–117
*[to be evaluated]*

*Franck, Egon* a. *Carola Jungwirth*: Die Governance von Open Source Projekten; in: Zeitschrift für Betriebswirtschaft, 73 (2003), No. 5 (Ergänzungsheft), pp. 1 –21
*next action, aus buchtala [bibliographic data have be verified]*

*Franky, María Consuelo*: Agile Management and Development of Software Projects based on Collaborative Environments; in: SIGSOFT Software Engineering Notes, 36 May (2011), No. 3, pp. 1–6 ⟨URL: `http://doi.acm.org/10.1145/1968587.1968605`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Free Software Foundation*: GNU General Public License, version 2; 1991 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://www.gnu.org/licenses/gpl-2.0.html`⟩ – reference download: 2013-02-05

*The GPL-2.0 license, using a strict copyleft. The first release was published in 1989. One of the main authors is probably Richard M. Stallman itself, even if today the Free Software Foundation is the copyright holder.*

*Free Software Foundation*: GNU Library General Public License [version 2.0]; 1991 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://www.gnu.org/licenses/old-licenses/lgpl-2.0.html`⟩ – reference download: 2013-03-25

*The LGPL-2.0 – the first GNU license with weak copyleft*

*Free Software Foundation*: GNU Lesser General Public License [Version 2.1]; 1999 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://www.gnu.org/licenses/lgpl-2.1.html`⟩ – reference download: 2013-03-06

*The LGPL-2.1 license – as the counterpart of the GPL-2.0 – offers a weak copyleft effect.*

*Free Software Foundation*: GNU General Public License [version 3]; 2007 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://www.gnu.org/licenses/gpl.html`⟩ – reference download: 2013-03-06

*The GPL-3.0 license – the current GNU license using a strong copyleft effect*

*Free Software Foundation*: GNU Lesser General Public License [version 3]; 2007 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://www.gnu.org/copyleft/lesser.html`⟩ – reference download: 2013-03-06

*The LGPL-3.0 license – the current GNU license using a weak copyleft effect*

*Free Software Foundation*: GNU Operating System[:] Licenses; 2011, FreeWeb/HTML ⟨URL: `http://www.gnu.org/licenses/`⟩ – reference download: 2013-03-25

*The main entry of the license specific section of the GNU operating system homepage*

*Free Software Foundation*: Various Licenses and Comments about Them; 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://www.gnu.org/licenses/license-list.html`⟩ – reference download: 2013-02-08

*The GNU list of software licenses, classified by the categories 'GPL-Compatible Free Software Licenses', 'GPL-Incompatible Free Software Licenses', and 'Nonfree Software Licenses'*

*Friedman, Batya et al.*: Development of a Privacy Addendum for Open Source Licenses: Value Sensitive Design in Industry; in: *Paul Dourish* a. *Adrian Friday, editors:* UbiComp 2006: Ubiquitous Computing; [Proceedings of the] 8th International Conference, UbiComp 2006 Orange County, CA, USA, September 17-21, 2006; Berlin, Heidelberg, a. New York: Springer, 2006 (= Lecture Notes in Computer Science, [Vol./No.] 4206), BibWeb/PDF, ISBN 978–3–540–39634–5, pp. 194–211

*[to be evaluated]*

*Fujita, Kunihiko* a. *Yasuyuki Tsukada*: An Analysis of Interoperability between Licenses; in: Proceedings of the Tenth Annual ACM Workshop on Digital Rights Management; New York, NY, USA: ACM, 2010 (= DRM '10) ⟨URL: `http://doi.acm.org/10.1145/1866870.1866884`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0091–9, pp. 61–72

*[to be evaluated]*

*Funk, Axel* a. *Georg Zeitfang*: Die GNU General Public License, Version3; in: CR [Computer und Recht], (2007), pp. 617 – 624

*[bibliographic data have be verified]*

*Funk, Axel* a. *Gregor Zeitfang*: Die GNU General Public License, Version 3; in: Computer und Recht, 23 (2007), No. 10, pp. 617–624, BibWeb/Copy

*[evaluate copy]*

*Fuse Source Team, The*: How to Use Open Source Integration Software Safely in the Enterprise. Analysis of potential risks and how to protect your IT environment; October 2010, FreeWeb/PDF [ File received by a promotion campaign of itwhitepapers.com. The article

refers to http:// www. fusesource. com/. But we didn't retrieve the paper there. ] ⟨URL: `http://www.itwhitepapers.com/?option=com_categoryreport\&task=viewabstract\&pathway=no\&autodn=1\&title=14770\&crv=0\&src=5053\&ctg=410\&cmp=3732\&yld=0\&pi=1628115`⟩ – reference download: 2011-08-24

*[tbd. eval PDF]*

*Gallini, Nancy* a. *Suzanne Scotchmer*: Intellectual Property: When Is It the Best Incentive System? in: NBER Innovation Policy & the Economy, 2 (2002), No. 1, pp. 51–77

*next action, aus buchtala [bibliographic data have be verified]*

*Gambardella, Alfonso* a. *Bronwyn H. Hall*: Proprietary versus public domain licensing of software and research products; in: RP, 35 (2006), No. 6, pp. 875–892 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S0048733306000643`⟩ – reference download: 2012-02-09, BibWeb/PDF

*[to be evaluated]*

*Gandal, Neil* a. *Chaim Fershtman*: Open Source Projectsd: Output per Contributor and Restrictive Licensing; o.O.: ???, 2004 (= (= CEPR Working Paper 2650))

*next action, aus buchtala [bibliographic data have be verified]*

*Gassmann, Oliver* a. *Martin A. Bader*: Patentmanagement: Innovationen erfolgreich nutzen und schützen; Berlin: ???, 2005

*next action, aus buchtala - könnte was über zwangsweise freigabe / lizenzierung enthalten [bibliographic data have be verified]*

*Geese, Elmar*: Innovation und freie Software; In *Drossou, Krempl, a. Poltmann*: Die wunderbare Wissensvermehrung, 2006, pp. 77–84, Print

*The article states that we are living in a time of monopolizing knowledge. It is established by patents, drm and laws. By supporting the privatization of knowlegde politics undermine their aim to support innovation. Positive counterexamples for a free culture are the maintenance of Staroffice/OpenOffice or Mozilla.*

*Gehring, Robert A.* a. *Bernd Lutterbeck, editors*: Open Source Jahrbuch 2004. Zwischen Softwareentwicklung und Gesellschaftsmodell; Berlin: Lehmanns Media, 2004 ⟨URL: `http://www.opensourcejahrbuch.de/download/jb2004/OpenSourceJahrbuch2004.pdf`⟩ – reference download: 2011-08-29, Print & FreeWeb/PDF, ISBN 3–936427–78–X

*First volume of the famous German written row 'Open Source Annual'. Offers articles on heterogeneous aspects of Open Source Software.*

*Gerber, Aurona*, *Onkgopotse Molefe*, a. *Alta van der Merwe*: Documenting Open Source Migration Processes for Re-use; in: Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists; New York, NY, USA: ACM, 2010 (= SAICSIT '10) ⟨URL: `http://doi.acm.org/10.1145/1899503.1899512`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–60558–950–3, pp. 75–85

*[to be evaluated]*

*Gerlach, Carsten*: Praxisprobleme der Open-Source-Lizenzierung; in: CR [Computer und Recht], (2006), pp. 649 – 654

*[bibliographic data have be verified]*

*German, Daniel M.*: Using software distributions to understand the relationship among free and open source software projects; in: Proceedings of the Fourth International Workshop on Mining Software Repositories; Washington, DC, USA: IEEE Computer Society, 2007 (= MSR '07) ⟨URL: `http://dl.acm.org/citation.cfm?id=1268983.1269038`⟩, BibWeb/PDF, ISBN 0–7695–2950–X, p. 24

*[to be evaluated]*

*German, Daniel M.* a. *Jesús M. González-Barahona*: An Empirical Study of the Reuse of

Software Licensed under the GNU General Public License; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 185–198, BibWeb/PDF
*[tbd: evaluate copy]*

*German, Daniel M.* a. *Ahmed E. Hassan*: License Integration Patterns: Addressing License Mismatches in Component-Based Development; in: Proceedings of the 31st International Conference on Software Engineering; Washington, DC, USA: IEEE Computer Society, 2009 (= ICSE '09) ⟨URL: `http://dx.doi.org/10.1109/ICSE.2009.5070520`⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–4244–3453–4, pp. 188–198
*[to be evaluated]*

*German, Daniel M.*, *Yuki Manabe*, a. *Katsuro Inoue*: A Sentence-Matching Method for Automatic License Identification of Source Code Files; in: Proceedings of the IEEE/ACM International Conference on Automated Software Engineering; New York, NY, USA: ACM, 2010 (= ASE '10) ⟨URL: `http://doi.acm.org/10.1145/1858996.1859088`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0116–9, pp. 437–446
*[to be evaluated]*

*German, Daniel M.*, *Jens H. Webber*, a. *Massimiliano Di Penta*: Lawful Software Engineering; in: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research; New York, NY, USA: ACM, 2010 (= FoSER '10) ⟨URL: `http://doi.acm.org/10.1145/1882362.1882390`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0427–6, pp. 129–132
*[to be evaluated]*

*Gilroy, Bernard Michael* a. *Tobias Volpert*: Die Funktionen eines Patentsystems und ihre Bedeutung für Unternehmensausgründungen aus Hochschulen; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 21–39, Print
*This article outlines the two social functions of a patent system: Firstly it shall evoke innovations by granting an exclusive right to use the invented results. Secondly it shall inform the scientific society by requiring the publication of new data. Finally the article argues that publishing information also evokes innovations because it's the base of succedding research.*

*Gobeille, Robert*: The FOSSology Project; in: Proceedings of the 2008 international working conference on Mining software repositories; New York, NY, USA: ACM, 2008 (= MSR '08) ⟨URL: `http://doi.acm.org/10.1145/1370750.1370763`⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–60558–024–1, pp. 47–50
*[to be evaluated]*

*Godfrey, Michael* a. *Qiang Tu*: Growth, Evolution, and Structural Change in Open Source Software; in: Proceedings of the 4th International Workshop on Principles of Software Evolution; New York, NY, USA: ACM, 2001 (= IWPSE '01) ⟨URL: `http://doi.acm.org/10.1145/602461.602482`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 1–58113–508–4, pp. 103–106
*[to be evaluated]*

*Goeminne, Mathieu* a. *Tom Mens*: A Framework for Analysing and Visualising Open Source Software Ecosystems; in: Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE); New York, NY, USA: ACM, 2010 (= IWPSE-EVOL '10) ⟨URL: `http://doi.acm.org/10.1145/1862372.1862384`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0128–2, pp. 42–47
*[to be evaluated]*

*Goldmann, R.* a. *R.P. Gabriel*: Innovation Happens Elsewhere. Opne Source as Business Strategy; San Francisco: Elsevier, 2005
*aus Simone Käs [bibliographic data have be verified]*

*Gomulkiewicz, Robert W.*: De-Bugging Open Source Software Licensing; in: University of Pittsburgh Law Review, 64 (2002), pp. 75–99, BibWeb/PDF
*[to be evaluated]*

*González-Barahona, Jesús M.*, *Joaquín Seoane Pascual*, a. *Gregorio Robles*: Introduction to Free Software; coordinated by Jordi Mas Hernández and David Megías Jiménez; Oberta de Catalunya: Free Technology Academy, 2009 ⟨URL: `http://www.ftacademy.org/materials/fsm/1#1`⟩ – reference download: 2012-01-20, FreeWeb/PDF
*eval copy*

*Goode, Sigi*: Something for nothing: management rejection of open source software in Australia's top firms; in: Information & Management, 669–681 (2005), p. 42, BibWeb/PDF
*[to be evaluated]*

*Gordon, Thomas F.*: Analyzing Open Source License Compatibility Issues with Carneades; in: Proceedings of the 13th International Conference on Artificial Intelligence and Law; New York, NY, USA: ACM, 2011 (= ICAIL '11) ⟨URL: `http://doi.acm.org/10.1145/2018358.2018364`⟩ – reference download: 2011-12-29, BIbWeb/PDF, ISBN 978–1–4503–0755–0, pp. 51–55
*[to be evaluated]*

*Grassmuck, Volker*: Open Source - Betriebssysteme für eine freiheitliche Gesellschaft; ⟨URL: `http://www.waste.informatik.hu-berlin.de/Grassmuck/Texts/OSS-Tutzing-5-00.html`⟩
*[bibliographic data have be verified]*

*Grassmuck, Volker*: Freie Software. Geschichte, Dynamiken und gesellschaftliche Bezüge; Berlin, 2000
*[next action, aus widmer] [bibliographic data have be verified]*

*Grassmuck, Volker*: Lizenzmodelle; 2000 ⟨URL: `http://www.mikro.org/Events/OS/text/lizenzen.html`⟩
*[bibliographic data have be verified]*

*Grassmuck, Volker*: Freie Software. Zwischen Privat- und Gemeineigentum; Themen und Materialien; Bonn: Bundeszentrale für politische Bildung, 2002, Print, ISBN 3–89331–432–6
*German standard work. The book was written in that time when Open Source seemed to be strange. It still offers a reasonable descriptions although the set of licences has grown in the mean time.*

*Green, Collin et al.*: Leveraging Open-Source Software in the Design and Development Process; in: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems; New York, NY, USA: ACM, 2009 (= CHI EA '09) ⟨URL: `http://doi.acm.org/10.1145/1520340.1520433`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–60558–247–4, pp. 3061–3074
*[to be evaluated]*

*Green, Eric Lee*: Economics of Open Source Software; 1998 ⟨URL: `http://badtux.org/home/eric/editorial/economics.php`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Green, Lisa* a. *Heather Meeker*: Open Software Licenses: Part II; in: Intellectual Property Strategist 10 (1999)
*[aus widmer, next action] [bibliographic data have be verified]*

*Greve, Georg C. F.*: Geschichte und Philosophie des GNU Projekts; ⟨URL: `http://www.gnu.org/philosophy/greeve-clown.html`⟩
*[aus Oberhem] [bibliographic data have be verified]*

*Grodzinsky, F. S.* a. *M. C. Bottis*: Private Use as Fair Use: Is it Fair? in: SIGCAS, 37 November (2007), No. 2, pp. 11–24 ⟨URL: http://doi.acm.org/10.1145/1327325.1327326⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Gräfe, Gernot*: Open-Source-Software und Open-Source-Portale - Potentiale für die Softwareentwicklung in Hochschulen und den Ergebnistransfer in die Praxis; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 55–72, Print
*The article lists some opportunities of university specific open source software portals, as the support of the software development, a better reuse of the development results or a better transfer of the theoretical results into the praxis.*

*Grützmacher, Malte*: Open-Source-Software - die GNU General Public License / Lizenzbestimmungen im Umfeld des neuen Schuld- und Urhebervertragsrechts; in: ITRB (IT-Rechtsberater), (2002), pp. 84ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Grützmacher, Malte*: Open Source Software - BSD Copyright and Apache Software. Copyright statt Copyleft; in: ITRB, o.A. (2006), No. 5, pp. 1008–112, Copy
*fernleihe*

*Guibault, Lucie* a. *Ot van Daalen*: Unravelling the Myth around Open Source Licenses. An Anaysis from A Dutch and European Law Perspective; The Hague: T. M. C. Asser Press, 2006 (= IT & Law, [Vol./No.] 8), Print, ISBN 978–90–6704–214–7
*The book describes the Open Source idea from the viewpoint of the Dutch legal system. It ends in five recommendations for enforcing the clarity of OS licenses and their usage. Except for the title the myth of Open Source is not explicitly mentioned or discussed.*

*Gull, Daniel*: Valuation of Discount Options in Software License Agreements; in: BISE, 4 (2011), pp. 221–230 ⟨URL: http://dx.doi.org/10.1007/s12599-011-0170-8⟩ – reference download: 2012-02-09, BibWeb/PDF
*[to be evaluated]*

*Gurbani, Vijay K.*, *Anita Garvert*, a. *James D. Herbsleb*: Managing a Corporate Open Source Software Asset; in: Commununications of the ACM, 53 February (2010), No. 2, pp. 155–159 ⟨URL: http://doi.acm.org/10.1145/1646353.1646392⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Gutsche, Jörg*: Ökonomische Analyse offener Software; Mannheim: Universität Mannheim, 2006, BibWeb/PDF
*[tbd: evaluate copy]*

*Gutwin, Carl*, *Reagan Penner*, a. *Kevin Schneider*: Group Awareness in Distributed Software Development; in: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work; New York, NY, USA: ACM, 2004 (= CSCW '04) ⟨URL: http://doi.acm.org/10.1145/1031607.1031621⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–810–5, pp. 72–81
*[to be evaluated]*

*Haase, H.*: Die Patentierbarkeit von Computersoftware; Hamburg, 2003
*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Haddad, I.*: Adopting an Open Source Approach to Software Development, Distribution, and Licensing; in: Enterprise Open Source Magazine, (2007) ⟨URL: http://opensource.sys-con.com/read/318776.htm⟩
*[bibliographic data have be verified]*

*Haddad, Ibrahim*: Open-Source Compliance; in: Linux Journal, 185 September (2009), p. 5:1 ⟨URL: http://dl.acm.org/citation.cfm?id=1610564.1610569⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Hahn, Robert, editor*: Government Policy toward Open Source Software; AEI-Brooking Joint Centre (Org.) 2002
*next action, aus buchtala [bibliographic data have be verified]*

*Hamerly, Jim*, *Tom Paquin*, a. *Susan Walton*: Freeing the Source: The Story of Mozilla; In *DiBona, Ockman, a. Stone*: Open Sources, 1999, pp. 197–206
*next action, aus buchtala [bibliographic data have be verified]*

*Hammouda, Imed et al.*: Open source legality patterns: architectural design decisions motivated by legal concerns; in: Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments; New York, NY, USA: ACM, 2010 (= MindTrek '10) ⟨URL: http://doi.acm.org/10.1145/1930488.1930533⟩ – reference download: 2012-01-06, BibWeb/PDF, ISBN 978–1–4503–0011–7, pp. 207–214
*[to be evaluated]*

*Hardaway, Donald E.*: Sharing Research in the 21st Century: Borrowing a Page from Open Source Software; in: Communications of the ACM, 48 August (2005), No. 8, pp. 125–128 ⟨URL: http://doi.acm.org/10.1145/1076211.1076216⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Hars, Alexander* a. *Shaosong Ou*: Working for free? Motivations for participating in open source projects; in: International Journal of Electronic Commerce, 6 (2002), No. 3, pp. 25–39
*[bibliographic data have be verified]*

*Hauge, Oyvind et al.*: An Empirical Study on Selection of Open Source Software - Preliminary results; In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009, pp. 42–47 ⟨URL: http://dx.doi.org/10.1109/FLOSS.2009.5071359⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Hauge, Oyvind* a. *Sven Ziemer*: Providing Commercial Open Source Software: Lessons Learned; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 70–82, BibWeb/PDF
*[tbd: evaluate copy]*

*Hawkins, R. E.*: The Economics of Open Source Software in a Competitive Firm: Why Give It Away For Free? in: Netnomics, 6 (2004), pp. 103–117
*[bibliographic data have be verified]*

*Heap, Nicholas*: OSI-Referenzmodell ohne Geheimnis; translated by G & U, Flensburg; Hannover: Heise, 1994, Print, ISBN 3–88229–045–5
*In the most cases 'OSI' refers the 'Open Systems Interconnection Model', not the 'Open Source Initiative'. Here a German written book explaining the elder meaning of OSI as a model of network layers.*

*Hecker, F.*: Setting up Shop: The Business of Open Source Software; in: IEEE Software, Jan/Feb (1999), pp. 46–61
*[bibliographic data have be verified]*

*Heffan, Ira V.*: Copyleft: Licensing Collaborative Works in the Digital Age; in: Stanford Law Review, 1997 (49), pp. 1487–1521 ⟨URL: http://www.jstor.org/stable/1229351⟩ – reference download: 2012-02-09, BibWeb/PDF
*[to be evaluated]*

*Heise online*: Deutsches Gericht bestätigt Wirksamkeit der GPL; 2004 ⟨URL: `http://www.heise,de/newsticker/meldung/49377`⟩
*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Heise online*: Der Streit um Softwarepateht; 2007 ⟨URL: `http://www.heise,de/ct/hintergrund/meldung/61230`⟩
*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Hemel, Armijn et al.*: Finding Software License Violations Through Binary Code Clone Detection; in: Proceedings of the 8th Working Conference on Mining Software Repositories; New York, NY, USA: ACM, 2011 (= MSR '11) ⟨URL: `http://doi.acm.org/10.1145/1985441.1985453`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0574–7, pp. 63–72
*[to be evaluated]*

*Henkel, Joachim*: Software development in embedded Linux; In *Uhr, Esswein, a. Schoop*: Wirtschaftsinformatik 2003 / Band II, 2003, pp. 81–99
*next action, aus buchtala [bibliographic data have be verified]*

*Henkel, Joachim*: The Jukebox Mode of Innovation; A Model of Commercial Open Source Development; 2004 ⟨URL: `http://opensource.mit.edu/papers/henkel.pdf`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Henkel, Joachim*: Open source software from commercial firms; Tools, complements, and collective invention; in: Zeitschrift für Betriebswirtschaft, 4/2004 (2004), No. 4 Ergänzungsheft, pp. 1–23
*next action, aus buchtala[bibliographic data have be verified]*

*Henkel, Joachim*: Patterns of Free Revealing; Balancing Code Sharing and Protection in Commercial Open Source Development; 2004 ⟨URL: `http://opensource.mit.edu/papers/henkel2.pdf`⟩
*[bibliographic data have be verified]*

*Henkel, Joachim*: Offene Innovationsprozesse. Die kommerzielle Entwicklung von Open-Source-Software; Wiesbaden: Deutscher Universitäts-Verlag, 2007 (= Gabler Edition Wissenschaft), BibWeb/PDF, ISBN 978–3–8350–0978–3
*[tbd: evaluate copy]*

*Herraiz, Israel et al.*: The Processes of Joining in Global Distributed Software Projects; in: Proceedings of the 2006 International Workshop on Global Software Development for the Practitioner; New York, NY, USA: ACM, 2006 (= GSD '06) ⟨URL: `http://doi.acm.org/10.1145/1138506.1138513`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–59593–404–9, pp. 27–33
*[to be evaluated]*

*Herraiz, Israel*, *Gregorio Robles*, a. *Jesus M. Gonzalez-Barahona*: Towards Predictor Models for large Libre Software Projects; in: Proceedings of the 2005 Workshop on Predictor Models in Software Engineering; New York, NY, USA: ACM, 2005 (= PROMISE '05) ⟨URL: `http://doi.acm.org/10.1145/1082983.1083168`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN –159593–125–2, pp. 1–6
*[to be evaluated]*

*Herraiz, Israel*, *Gregorio Robles*, a. *Jesus M. Gonzalez-Barahona*: Research Friendly Software Repositories; in: Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution (IWPSE) and Software Evolution (Evol) Workshops; New York, NY, USA: ACM, 2009 (= IWPSE-Evol '09) ⟨URL: `http://doi.acm.org/10.1145/1595808.1595814`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–678–6, pp. 19–24
*[to be evaluated]*

*Hertel, Guido*, *Sven Niedner*, a. *Stefanie Herrmann*: Motivation of Software Developers in Open Source Projects: An Internet-based Survey of Contributors to the Linux Kernel; in: Research Policy, 32 (2003), pp. 1159–1177
*next action, aus buchtala [bibliographic data have be verified]*

*Hill, Benjamin Mako*: Samir Chopra, Scott D. Dexter, Decoding Liberation: The Promise of Free and Open Source Software; in: Minds Mach. 18 June (2008), pp. 297–299 ⟨URL: `http://dl.acm.org/citation.cfm?id=1375424.1375428`⟩, ISSN 0924–6495
*Rezension: evaluate the referred book [bibliographic data have be verified]*

*Hislop, Gregory W. et al.*: Using Open Source sSoftware to Engage Students in Computer Science Education; in: Proceedings of the 40th ACM technical symposium on Computer science education; New York, NY, USA: ACM, 2009 (= SIGCSE '09) ⟨URL: `http://doi.acm.org/10.1145/1508865.1508915`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–183–5, pp. 134–135
*[to be evaluated]*

*Hissam, Scott A. et al., editors*: Open Source Systems: Grounding Research; 7th IFIP WG 2.13 International Conference, OSS 2011; (= IFIP Advances in Information and Communication Technology, [Vol./No.] 365) Heidelberg, Dordrecht, London u. NewYork: Springer, 2011, BibWeb/PDF, ISBN 978–3–642–24418–6
*eval article copies*

*Hoeren, Thomas*: Anmerkungen zum Urteil vom 19.5.2004 des LG München I zur Wirksamkeit einer GPL-Lizenz; in: CR [Computer und Recht], (2004), pp. 776–778
*[aus oberhem] [bibliographic data have be verified]*

*Hoeren, Thomas*: Internetrecht; April 2011, Web/PDF? ⟨URL: `http://www.uni-muenster.de/Jura.itm/hoeren/lehre/materialien`⟩
*Überblick über das gesamte Internetrecht, OSS eher wenig [bibliographic data have be verified]*

*Hofmann, Susanne*, *Sven Pfeiffer*, a. *Urs Walter*: Open Source School. Neue Synergien zwischen Schule und Kiez in Gropiusstadt. Architektur als sozialer Katalysator; Berlin, 2010, BibWeb/PDF ⟨URL: `http://opus.kobv.de/tuberlin/volltexte/2010/2841/pdf/9783798322738_content.pdf`⟩ – reference download: 2011-07-30
*Although using a seducing title the pdf file doesn't offer anything concerning 'Open Source Software': topic is a special kind of managing architecture of buildings.*

*Horne, Natasha*: Open Source Software Licensing: Using Copyright Law to Encourage Free Use; in: Georgia State University Law Review, (2001), pp. 863–891
*[aus widmer] [bibliographic data have be verified]*

*Horns, A.*: Der Patentschutz für software bezogene Erfindungen im Verhätnis zur 'Open Source'-Software; in: Zeitschrift für Rechtsinformatik, (2000) ⟨URL: `http://www.jurpc.de/aufsatz/2000223.htm`⟩
*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Howland, John E.*: Software Freedom, Open Software and the Undergraduate Computer Science Curriculum; in: JCSC, 15 March (2000), No. 3, pp. 293–301 ⟨URL: `http://dl.acm.org/citation.cfm?id=1852563.1852604`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Howland, John E.*: Managing Computer Science Laboratories Using Open Software; in: JCSC, 16 March (2001), No. 3, pp. 117–126 ⟨URL: `http://dl.acm.org/citation.cfm?id=374685.374726`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Hubbard, Jordan*: Open Source to the Core; in: Queue, 2 (2004), pp. 24–31 ⟨URL: `http://doi.acm.org/10.1145/1005062.1005064`⟩, BibWeb/PDF
*[to be evaluated]*

*ifross*: FAQ; 2011, FreeWeb/HTML ⟨URL: `http://www.ifross.org/faq-haeufig-gestellte-fragen`⟩ – reference download: 2011-09-05
*This page presents quintessences concerning the topic 'Open Source' - in form of a FAQ list*

*ifross*: Ziele, Aufgaben, Geschichte; 2011, FreeWeb/HTML ⟨URL: `http://www.ifross.org/node/16`⟩ – reference download: 2011-09-05
*This page describes the targets and the history of ifross, the 'Institut für Rechtsfragen der Freien und Open Source Software'. It was founded in 2000 as a private institute which shall accompany the phenomenon 'free software' from the viewpoint of (German) lawyers.*

*ifross*: License Center; 2011 [n.y.], FreeWeb/HTML ⟨URL: `http://www.ifross.org/ifross_html/lizenzcenter-en.html`⟩ – reference download: 2013-02-26
*This page lists many more (Open Source) Licenses than the OSI itself. It's classifying the Open Source Licenses in those without copyleft-effect, in those with strict copyleft-effect and in those with restricted copyleft-effect - one of the most sophisticated and elaborated considerations!*

*ifross et al.*: Die GPL kommentiert und erklärt; Beijing, Cambridge, Farnham [etc ..]: O'Reilly, 2005, Print, ISBN 3–89721–389–3
*This book explains the legal implications and meaning of each GPL section - including the meaning of 'derivative work'*

*Imhorst, Christian*: Die Anarchie der Hacker. Richard Stallman und die Freie-Software-Bewegung; Marburg: Tectum Verlag, 2004, Print, ISBN 3–8288–8769–4
*This book tries - a little willfully - to tell the story of Open Source as a consequence of the American kind of anarchy: basically it mentions the two branches of this history, the personal computer on the one hand (lateron allegedly monopolized by Microsoft) and the university tradition of free collaboration and exchange.*

*Izurieta, Clemente* a. *James Bieman*: The Evolution of FreeBSD and Linux; in: Proceedings of the 2006 ACM/IEEE international Symposium on Empirical Software Engineering; New York, NY, USA: ACM, 2006 (= ISESE '06) ⟨URL: `http://doi.acm.org/10.1145/1159733.1159765`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 1–59593–218–6, pp. 204–211
*[to be evaluated]*

*Jackson, Darla W.*: Thinking about Technology . . . Watson, Answer Me This: Will You Make Librarians Obsolete or Can I Use Free and Open Source Software and Cloud Computing to Ensure a Bright Future? in: Law Library Journal, 103 (2011), No. 3, pp. 497–504, BibWeb/PDF
*[to be evaluated]*

*Jacobs, Stephen*, *Clif Kussmaul*, a. *Mihaela Sabin*: Free and Open Source Software in Computing Education; in: Proceedings of the 2011 Conference on Information Technology Education; New York, NY, USA: ACM, 2011 (= SIGITE '11) ⟨URL: `http://doi.acm.org/10.1145/2047594.2047606`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–1017–8, pp. 41–42
*[to be evaluated]*

*Jaeger, Till*: Copyright oder Copyleft; in: Computerwoche Spezial, 27 (2000), No. 4, p. 36 ⟨URL: `http://www.ifross,de/ifross_html/art6.html`⟩
*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Jaeger, Till*: GPL und Haftung: Ohne Verantwortung? in: Linux-Magazin, (2000), No. 5, pp. 134ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Jaeger, Till*: Die GPL kommentiert und erklärt; hrsg; v. ifross; Köln, 2005
*aus Koeglin2007a [bibliographic data have be verified]*

*Jaeger, Till* a. *Axel Metzger*: Open Source Software und deutsches Urheberrecht; in: GRUR Int. (1999), pp. 839ff
*aus Koeglin2007a[bibliographic data have be verified]*

*Jaeger, Till* a. *Axel Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 1st edition. München: Verlag C.H. Beck, 2002, Print, ISBN 3406484026
*This is the first edition of the most important German standard work concerning Open Source Software Licenses. It deals with many important topics like liability, patents, or branding, And with respect to each of the mentioned licenses (or license clusters) it discusses already the rights and the duties of the software user - but regrettably not in form of a processable todo-list and not with distinguishing the different use cases of the software development process. But nevertheless: it's a very important groundwork!*

*Jaeger, Till* a. *Axel Metzger*: Open Content-Lizenzen nach deutschem Recht; in: MultiMedia und Recht, (2003), pp. 431ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Jaeger, Till* a. *Axel Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 2nd edition. München: Verlag C.H. Beck, 2006, Print, ISBN 3406538037
*This is the second strongly expanded edition of the most important German standard work concerning Open Source Software Licenses. Like ifross it classifies licenses as those with 'strict copyleft clauses', those with 'restricted copyleft-clauses' and those without any 'copyleft-clauses'. Beside other aspects it now discusses how to achieve acceptance of an Open Source licensing and analyzes the AGB side effects. And again it discusses the rights and the duties of the software user, at least with respect to the most import Open Source Licenses - but again regrettably not in form of a processable todo-list and not with distinguishing the different use cases of the software development process. But naturally: it remains a very important groundwork!*

*Jaeger, Till* a. *Axel Metzger*: Die neue Version 3 der GNU general Public License; in: GRUR (Gerwerblicher Rechtsschutz und Urheberrecht), (2008), pp. 130–137
*[bibliographic data have be verified]*

*Jaeger, Till* a. *Axel Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 3rd edition. München: Verlag C.H. Beck, 2011, Print
*This is the third current edition of the most important German standard work concerning Open Source Software Licenses. It retains the 'established structure of the chapters', adds new licenses and discusses - beside many other aspects - the compatibility respectively combinability of different licenses. Hence this work again specifies the rights and the duties of the software user, at least with respect to the most important licenses - but again regrettably not in form of a processable todo-list and not with distinguishing the different use cases of the software development process. But it retains the most important groundwork!*

*Jaeger, Till* a. *Carsten Schulz*: Gutachten zu ausgewählten rechtlichen Aspekten der Open Source Software - im Rahmen des Projektes 'NOW - Nutzung des Open Source-Konzepts in Wirtschaft und Industrie'; Feb 2005
*[bibliographic data have be verified]*

*Janamanchi, Balaji et al.*: The State and Profile of Open Source Software Projects in health and medical informatics; in: International Journal of Medical Informatics, 78 (2009), No. 7, pp. 457–472 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S1386505609000318`⟩ – reference download: 2012-02-09, BibWeb/PDF
*[to be evaluated]*

*Jansson, Kurt*, *Patrick Danowski*, a. *Jakob Voss*: Wikipedia: Kreative Anarchie für den freien Informations- und Wissensaustausch; In *Drossou, Krempl, a. Poltmann*: Die wunderbare Wissensvermehrung, 2006, pp. 159–167, Print

*The article describes shortly the character of 'wikipedia', its' link to the encyclopedists around Diderot and the parallelism of Wikipedia and the Open Source development process,*

*Jendroska, Dirk*: Arbeitsgestaltung in der Softwareentwicklung: Ein empirischer Vergleich subjektiver Arbeitsmerkmale in proprietären und Open Source Softwareprojekten; Dissertation; Münster: Philosophischen Fakultät der Westfälischen Wilhelms Universität zu Münster, 2010, BibWeb/PDF
*[tbd. eval ecopie]*

*Johnson, Justin P.*: Open Source Software: Private Provision of a Public Good; in: Journal of Economics & Management Strategy, 11 (2002), No. 4, pp. 637–663
*next action, aus buchtala [bibliographic data have be verified]*

*Johnson, Michael K.*: Licenses and Copyright; in: Linux Journal, 29 September (1996), p. 3:1 ⟨URL: `http://dl.acm.org/citation.cfm?id=326350.326353`⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Johnson-Eilola, Johndan*: Open Source Basics: Definitions, Models, and Questions; in: Proceedings of the 20th Annual International Conference on Computer Documentation; New York, NY, USA: ACM, 2002 (= SIGDOC '02) ⟨URL: `http://doi.acm.org/10.1145/584955.584967`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–543–2, pp. 79–83
*[to be evaluated]*

*Johri, Aditya*, *Oded Nov*, a. *Raktim Mitra*: "Cool" or "Monster"? Company Takeovers and Their Effect on Open Source Community Participation; in: Proceedings of the 2011 iConference; New York, NY, USA: ACM, 2011 (= iConference '11) ⟨URL: `http://doi.acm.org/10.1145/1940761.1940806`⟩, ISBN 978–1–4503–0121–3, pp. 327–331
*[to be evaluated]*

*Jones, Paul*: Open (Source)ing the Doors for Contributor-run Digital Libraries; in: Communications of the ACM, 44 May (2001), No. 1, pp. 45–46 ⟨URL: `http://doi.acm.org/10.1145/374308.374337`⟩ – reference download: 2011-12-49, BibWeb/PDF
*[to be evaluated]*

*Karels, Michael J.*: Commercializing Open Source Software; in: Queue, 1 July/August (2003), pp. 46–55 ⟨URL: `http://doi.acm.org/10.1145/945074.945125`⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Karus, Siim* a. *Harald Gall*: A Study of Language Usage Evolution in Open Source Software; in: Proceedings of the 8th Working Conference on Mining Software Repositories; New York, NY, USA: ACM, 2011 (= MSR '11) ⟨URL: `http://doi.acm.org/10.1145/1985441.1985447`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–4503–0574–7, pp. 13–22
*[to be evaluated]*

*Kelty, Christopher M*: Free Software/Free Science; in: First Monday, 6 (2001), No. 12, p. o.A.
*[bibliographic data have be verified]*

*Kelty, Christopher M.*: Two Bits: The cultural Significance of Free Software; ???, 2008
*[Präsenz Dienstzimmer Welz Institut für Kulturanthropologie R.1.454: 02/AP 15990K29]*
*[bibliographic data have be verified]*

*Kennedy, D. M.*: A primer on open source licensing legal issues: copyright, copyleft and copyfuture; 2001 ⟨URL: `http://www.denniskennedy.com/opensourcedmk.pdf`⟩
*[bibliographic data have be verified]*

*Kern, W.* a. *F. Rammig*: Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte; Paderborn; in: C-LAB Report, 2 (2003), vielleicht Buch?

*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Kersken, Sasche*: Apache 2.2. Das umfassende Handbuch; 3rd, refreshed a. expanded edition; Bonn: Galileo Press, 2009, Print, ISBN 978–8362–1325–7
*An Apache compendium with more than 900 pages offering 2 pages concerning the history and the content of the Apache license.*

*Keuffel, Warren*: License Overload; in: Software Development, 14 (2006), No. 2, p. 56, Copy
*fernleihe*

*Keßler, Steffen* a. *Paul Alpar*: Customization of Open Source Software in Companies; 5th IFIP WG 2.13 International Conference on Open Source Systems, OSS 2009 Skövde, Sweden, June 3-6, 2009; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 129–142, BibWeb/PDF
*[tbd: evaluate copy]*

*Kienle, Holger M. et al.*: Intellectual Property Aspects of Web Publishing; in: Proceedings of the 22nd annual international conference on Design of communication: The engineering of quality documentation; New York, NY, USA: ACM, 2004 (= SIGDOC '04) ⟨URL: `http://doi.acm.org/10.1145/1026533.1026569`⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 1–58113–809–1, pp. 136–144
*[to be evaluated]*

*Kilamo, Terhi*: The Community Game: Learning Open Source Development Through Participatory Exercise; in: Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments; New York, NY, USA: ACM, 2010 (= MindTrek '10) ⟨URL: `http://doi.acm.org/10.1145/1930488.1930500`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0011–7, pp. 55–60
*[to be evaluated]*

*Kirschner, Bryan*: Building a Balanced Scorecard for Open Source Policy and Strategy: A Case Study of the Microsoft Experience; in: Proceedings of the 2nd International Conference on Theory and Practice of Electronic Governance; New York, NY, USA: ACM, 2008 (= ICEGOV '08) ⟨URL: `http://doi.acm.org/10.1145/1509096.1509142`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–386–0, pp. 226–231
*[to be evaluated]*

*Kitcat, Jason*: Source Availability and E-voting: An Advocate Recants; in: Communications of the ACM, 47 October (2004), No. 10, pp. 65–67 ⟨URL: `http://doi.acm.org/10.1145/1022594.1022625`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Klemm, Martin*: GPL Version 3.0; in: Innovation und internationale Rechtspraxis, o.A. (2009), pp. 363–381, Copy
*Fernleihkopie*

*Koch, Frank A.*: Urheber- und kartellrechtliche Aspekte der Nutzung von Open-Source-Software (I); in: CR [Computer und Recht], (2000), pp. 273ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Koch, Frank A.*: Urheber- und kartellrechtliche Aspekte der Nutzung von Open-Source-Software (II); in: CR [Computer und Recht], (2000), p. 333
*aus Koeglin2007a [bibliographic data have be verified]*

*Koch, Frank A.*: Probleme beim Wechsel zur neuen Version 3 der General Public License (Teil 1); in: ITRB (IT-Rechtsberater), (2007), pp. 261–263
*[aus oberhem] [bibliographic data have be verified]*

*Koch, Frank A.*: Probleme beim Wechsel zur neuen Version 3 der General Public License (Teil 2); in: ITRB (IT-Rechtsberater), (2007), pp. 285–288
*[aus oberhem] [bibliographic data have be verified]*

*Koeglin, Olaf*: Entfesseltes Wissen - Creative Commons und der Versuch, das GPL-Prinzip für jede Schöpfung anzuwenden; in: Linux-Magazin, (2003), No. 10, p. 70
*aus Koeglin2007a [bibliographic data have be verified]*

*Koeglin, Olaf*: Die Nutzung von Open Source Software unter neuen GPL Versionen nach der 'any later Version'-Klausel; in: CR [Computer und Recht], (2008), pp. 137–143
*[aus oberhem] [bibliographic data have be verified]*

*Koeglin, Olaf* a. *Axel Metzger*: Urheber- und Lizenzrecht im Bereich von Open-Source-Software; in Open Source Jahrbuch 2004; 2004, pp. 293ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Koenig, J.*: Seven Open Source Business Strategies for mCompetitive Advantage; in: IT Manager's Journal, (2004) ⟨URL: `http://management.itmanagersjournal.com/article?sid=04/05/10/2052216`⟩
*[bibliographic data have be verified]*

*Koglin, Olaf*: Opensourcerecht. Die urheber- und schuldrechtlichen Beziehungen zwischen Lizenzgeber und Lizenznehmer bei Open Source Software am Beispiel der General Public License (GPL); Frankfurt am Main: Peter Lang, 2007 (= Schriften zum Wirtschafts- und Medienrecht, Steuerrecht und Zivilprozeßrecht, [Vol./No.] 31), Print, ISBN 978–3–631–56308–3
*Thoroughly this book analyzes the validity of GPL with respect to the general German right, the German copyright ('Urheberrecht') and the German law of contract. Additionally it discusses all paragraphs of the GPL and their concret meaning as part of the German right. Other Open Source licenses are not outlined in the same manner.currently it is used more than twice as much as all the others together.*

*Kogut, B.* a. *A. Metiu*: Open Source Software Development and Distributed Innovation; in: Oxford Review of Economic Policy, 17 (2001), pp. 248–264
*next action, aus Käs [bibliographic data have be verified]*

*Koponen, Timo* a. *Virpi Hotti*: Open Source Software Maintenance Process Framework; In Proceedings of the Fifth Workshop on Open Source Software Engineering, 2005, pp. 4:1–4:5 ⟨URL: `http://doi.acm.org/10.1145/1082983.1083265`⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Kreutzer, Till*: Anmerkungen zum Urteil vom 19.5.2004 des LG München I zur Wirksamkeit einer GPL-Lizenz; in: MultiMedia und Recht, (2004), pp. 695–698
*[aus oberhem] [bibliographic data have be verified]*

*Kreutzer, Till*: Software und Spiele kopieren[:] Das Lizenzmodell entscheidet; In *Djordjevic et al.*: Urheberrecht im Alltag, 2008, pp. 29–33, Print
*Hints to the German 'Urheberrecht', which predetermines, that in Germany a back-up copy of software can't be forbidden by any licence and that all other types of copying software must explicitly be allowed by the connected licence.*

*Kreutzer, Till*: Software veröffentlichen[:] Wem gehören die Rechte? In *Djordjevic et al.*: Urheberrecht im Alltag, 2008, pp. 163–167, Print
*Specifies that 'Urheberrecht' and 'Verwertungsbefugnis' are initially linked in Germany. The right to use ('Verwertungsbefugnis') can be assigned to other, even generally: In Germany employed developers assign their right to determine the software use to their employers by signing the contract. But: the German 'Urheberrecht' doesn't include a protection of the embedded ideas.*

*Kreutzer, Till*: Softwarelizenzen - Beispiele[:] Und welche Lizenz nehme ich jetzt? In *Djordjevic et al.*: Urheberrecht im Alltag, 2008, pp. 176–179, Print
*Offers a case based analysis concerning the interests of the developer: Open Source licences can be used if he hasn't any further interests or if he wants to get back improvements of other*

*developer or if he wants to support his name; in other cases he should use a proprietary licence. The different Open Source licences are not analyzed in a deeper sense.*

*Krishnamurthya, Sandeep* a. *Arvind K. Tripathi*: Monetary donations to an open source software platform; in: Research Policy, 38 (2009), pp. 404–414
*[to be evaluated]*

*Krogstie, Birgit R.*: Power Through Brokering: Open Source Community Participation in Software Engineering Student Projects; in: Proceedings of the 30th International Conference on Software Engineering; New York, NY, USA: ACM, 2008 (= ICSE '08) ⟨URL: `http://doi.acm.org/10.1145/1368088.1368201`⟩ – reference download: 2011-12-29, BIb-Web/PDF, ISBN 978–1–60558–079–1, pp. 791–800
*[to be evaluated]*

*Kugler, Petra*: Coordinating Innovation: Evidence from Open Source Development; Dissertation; St. Gallen: University of. St. Gallen, 2005, Print
*This work focuses on the question, how informal organizations work, how they establish their hierarchies etc. Open Source communities are taken as examples. Roughly spoken their structures are volatily established on the base of domain and/or project knowledge. The book refers to Open Source licenses only and shortly as a method to found the free exchange of code. How to fulfill the license exactly is out of scope.*

*Kuhlen, Rainer*: Open Innovation: Teil einer nachhaltigen Wissensökonomie; In *Drossou, Krempl, a. Poltmann*: Die wunderbare Wissensvermehrung, 2006, pp. 12–23, Print
*[tbd: evaluate article in book]*

*Kumar, Vineet*, *Brett R. Gordon*, a. *Kannan Srinivasan*: Competitive Strategy for Open Source Software; in: MARKETING SCIENCE, 30 (2011), No. 6, pp. 1066–1078, BibWeb/PDF
*[bibliographic data have be verified]*

*Käs, Simone*: Rethinking industry practice. The emergence of openness in the embedded component industry; München: Pro BUSINESS, 2008, Print, ISBN 978–3–86805–256–5
*This book analyzes the change to open software developement by interviewing 'embeddedlinux'-companies. The result is that openness is required by the customers and that practicing openness evokes a process of learning. The book offers a short but tellingly survey of OSS and their licenses, although it doesn't offer a systematical review of the obligations established by the different licenses.*

*Lacy, Sarah*: Open Source: Now It's an Ecosystem; 2005 ⟨URL: `http://www.businessweek.com/technology/content/oct2005/tc2005103_0519_tc_218.htm`⟩
*[bibliographic data have be verified]*

*Lakhani, Karim R.* a. *Eric Hippel*: How Open Source software works: "Free" user-to-user assistance; 2002, MIT Sloan School of Management Working Paper
*next action, aus buchtala [bibliographic data have be verified]*

*Lakhani, Karim R.* a. *Robert G. Wolf*: Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects; (= MIT Sloan School of Management Working, Paper 4425-03), 2003
*next action, aus buchtala [bibliographic data have be verified]*

*Laroque, Chrstoph*, *Andre Döring*, a. *Thorsten Timm*: 'Give or Let Buy': Kritische Überlegungen eines Software-Ingeneurs zur Veröffentlichung von Software als Open-Source-Projekte; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 155–166, Print
*This article wants to address some criteria for and against the publication of source code, explicitly in an 'at least dimly' form. The article has acchieved its' goals. License challenges are not really mentioned.*

*Lavazza, Luigi et al.*: Predicting OSS Trustworthiness on the Basis of Elementary Code Sssessment; in: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement; New York, NY, USA: ACM, 2010 (= ESEM '10) ⟨URL: http://doi.acm.org/10.1145/1852786.1852834⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0039–1, pp. 36:1–36:4
*[to be evaluated]*

*Lawrie, Tony* a. *Cristina Gacek*: Issues of Dependability in Open Source Software Development; in: SIGSOFT Software Engineering Notes, 27 May (2002), No. 3, pp. 34–37 ⟨URL: http://doi.acm.org/10.1145/638574.638584⟩
*[to be evaluated]*

*Lee, Samuel*, *Nina Moisa*, a. *Marco Weiss. An Economic Analysis*: Open Source as a Signalling Device; Frankfurt a.M.: Goethe-University Frankfur/Main, 2003 (= Working Paper Series: Finance and Accounting, [Vol./No.] 102), BibWeb/PDF
*[tbd: evaluate copy]*

*Lee, Samuel*, *Nina Moisa*, a. *Marco Weiss*: Conditions for Open Source as a Signalling Device; Frankfurt a.M.: Goethe-University Frankfur/Main, 2004 (= Working Paper Series: Finance and Accounting), BibWeb/PDF
*[tbd: evaluate copy]*

*Lejeune, Mathias*: Rechtsprobleme bei der Lizenzierung von Open Source Software nach der GNU GPL; in: ITRB (IT-Rechtsberater), 1 (2003), pp. 10–12
*[aus widmer] [bibliographic data have be verified]*

*Lelli, Francesco* a. *Mehdi Jazayeri*: Community Support for Software Development in Small Groups: the Initial Steps; in: Proceedings of the 2nd international workshop on Social software engineering and applications; New York, NY, USA: ACM, 2009 (= SoSEA '09) ⟨URL: http://doi.acm.org/10.1145/1595836.1595840⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–682–3, pp. 15–22
*[to be evaluated]*

*Lemley, Mark A.* a. *Ziv Shafir*: Who Chooses Open-Source Software? in: University of Chicago Law Review, 78 (2011), pp. 139–163, BibWeb/PDF
*[to be evaluated]*

*Lenarcic, John* a. *Eric C. Mousset*: The Open Source Singularity: A Postmodernist View; in: Selected papers from conference on Computers and philosophy - Volume 37; Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003 (= CRPIT '03) ⟨URL: http://dl.acm.org/citation.cfm?id=1082145.1082157⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 1–920–68219–8, pp. 73–77
*[to be evaluated]*

*Lerner, Josh* a. *Jean Tirole*: The open source movement: Key research questions; in: European Economic Review, 45 (2001), pp. 819–826, BibWeb/PDF
*[to be evaluated]*

*Lerner, Josh* a. *Jean Tirole*: The Scope of Open Source Licensing; in: JLEO, 21 (2005), No. 1, pp. 20–56, BibWeb/PDF
*[to be evaluated]*

*Lerner, Joshua* a. *Jean Tirole*: Some simple economics of Open Source; in: Journal of Industrial Economics, 50 (2002), No. 2, pp. 197–234
*next action, aus buchtala, aus käs (untershciedlicher titel [bibliographic data have be verified]*

*Levy, S.*: Hackers; USA: Penguin, 2001

*Li, Yan*, *Chuan Hoo Tan*, a. *Hock Hai Teo*: Firm-Specificity and Organizational Learning-related Scale on Investment in Internal Human Capital for Open Source Software Adoption; in: Proceedings of the 2008 ACM SIGMIS CPR Conference on Computer Personnel Doctoral

Consortium and Research; New York, NY, USA: ACM, 2008 (= SIGMIS CPR '08) ⟨URL: http://doi.acm.org/10.1145/1355238.1355244⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–60558–069–2, pp. 22–29
*[to be evaluated]*

*Li, Yan et al.*: Motivating Open Source Software Developers: Influence of Transformational and Transactional Leaderships; in: Proceedings of the 2006 ACM SIGMIS CPR Conference on Computer Personnel Research: Forty Four Years of Computer Personnel Research: Achievements, Challenges & the Future; New York, NY, USA: ACM, 2006 (= SIGMIS CPR '06) ⟨URL: http://doi.acm.org/10.1145/1125170.1125182⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–59593–349–2, pp. 34–43
*[to be evaluated]*

*Li, Yan et al.*: Open Source Software Adoption: Motivations of Adopters and Amotivations of Non-adopters; in: SIGMIS Database, 42 May (2011), No. 2, pp. 76–94 ⟨URL: http://doi.acm.org/10.1145/1989098.1989103⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Li, Yung-Ming*, *Jhih-Hua Jhang-Li*, a. *Yen-Chun Liu*: Optimal Strategies of IT Consulting Firms: The Impact of License Fee and Open Source; in: Proceedings of the 10th International Conference on Electronic Commerce; New York, NY, USA: ACM, 2008 (= ICEC '08) ⟨URL: http://doi.acm.org/10.1145/1409540.1409594⟩ – reference download: 2011-12-29, ISBN 978–1–60558–075–3, pp. 40:1–40:7
*[to be evaluated]*

*Lin, Yi-Hsuan et al.*: Open Source Licenses and the Creative Commons Framework: License Selection and Comparison; in: JISE, 22 (2006), pp. 1–17, BibWeb/PDF
*[to be evaluated]*

*Lin, Yu-Wei* a. *Enrico Zini*: Free/libre open source software implementation in schools: Evidence from the field and implications for the future; in: Computers & Education, 50 (2008), No. 3, pp. 1092–1102 ⟨URL: http://www.sciencedirect.com/science/article/pii/S0360131506001722⟩, BibWeb/PDF, ISSN 0360–1315
*[to be evaluated]*

*Lindman, J.*, *M. Rossi*, a. *A. Puustell*: Matching Open Source Software Licenses with Corresponding Business Models; in: Software, IEEE, 28 july-aug. (2011), No. 4, pp. 31 –35, ISSN 0740–7459
*[bibliographic data have be verified]*

*Lindman, Juho*, *Juha-Pekka Juutilainen*, a. *Matti Rossi*: Beyond the Business Model: Incentives for Organizations to Publish Software Source Code; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 47–56, BibWeb/PDF
*[tbd: evaluate copy]*

*Lovett, Jayne*: Open Source - A Practical Solution; in: Proceedings of the 35th Annual ACM SIGUCCS Fall Conference; New York, NY, USA: ACM, 2007 (= SIGUCCS '07) ⟨URL: http://doi.acm.org/10.1145/1294046.1294099⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–59593–634–9, pp. 221–223
*[to be evaluated]*

*Lundell, Björn*, *Brian Lings*, a. *Edvin Lindqvist*: Open source in Swedish companies: where are we? in: Information Systems Journal, 20 (2010), p. 519–535, BibWeb/PDF
*[to be evaluated]*

*Lutterbeck, Bernd*, *Matthias Baerwolff*, a. *Robert A. Gehring, editors*: Open Source Jahrbuch 2006. Zwischen Softwareentwicklung und Gesellschaftsmodell; Berlin: Lehmanns Media, 2006 ⟨URL: http://www.opensourcejahrbuch.de/download/jb2006/OpenSourceJahrbuch2006_online.pdf⟩ – reference download: 2011-10-17, Print & Free-Web/PDF, ISBN 3–86541–135–5

*Third volume of the famous German written row 'Open Source Annual'. Offers articles on heterogeneous aspects of Open Source Software. [tbd: evaluate copy]*

*Lutterbeck, Bernd*, *Matthias Baerwolff*, a. *Robert A. Gehring, editors*: Open Source Jahrbuch 2007. Zwischen Softwareentwicklung und Gesellschaftsmodell; Berlin: Lehmanns Media, 2007 ⟨URL: `http://www.opensourcejahrbuch.de/download/jb2007/OpenSourceJahrbuch2007_online.pdf`⟩ – reference download: 2011-10-17, Print & FreeWeb/PDF, ISBN 978–3–86541–191–4

*Fourth volume of the famous German written row 'Open Source Annual'. Offers articles on heterogeneous aspects of Open Source Software. [tbd: evaluate copy]*

*Lutterbeck, Bernd*, *Matthias Baerwolff*, a. *Robert A. Gehring, editors*: Open Source Jahrbuch 2008. Zwischen Softwareentwicklung und Gesellschaftsmodell; Berlin: Lehmanns Media, 2008 ⟨URL: `http://www.opensourcejahrbuch.de/download/jb2008/osjb08.pdf`⟩ – reference download: 2011-10-17, Print & FreeWeb/PDF, ISBN 978–3–86541–271–3

*Fifth and last volume of the famous German written row 'Open Source Annual'. Offers articles on heterogeneous aspects of Open Source Software. [tbd: evaluate copy]*

*Lutterbeck, Bernd* a. *Robert A. Gehring Matthias Bärwolff, editors*: Open Source Jahrbuch; Berlin: Lehmanns Media, 2005 ⟨URL: `http://...mitaufnehmen..`⟩

*next action, aus buchtala [bibliographic data have be verified]*

*Maaß, C* a. *E. Schern*: Softwarepatente; in: Das Wirtschaftsstudium, 33 (2004), No. 10, pp. 1026–1028

*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Maaß, C.* a. *E. Schern*: Software-Lizenzierung; in: Das Wirtschaftsstudium, 34 (2005), No. 2, pp. 185–188

*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Maaß, Christian*: Zur Bedeutung des Urheber- und Patanterechts in der quelloffenen Softwareentwicklung; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 41–54, Print

*The article mentions that the European copryright protects the 'text', the patent the idea behind the text. In Europe it's possible to get a patent 'for' a piece of software only if the main claim concerns a technical invention. Nevertheless this disturbs the idea of Open Source Software. Finally the article hints to the patent pool established by the OSDL.*

*MacCormack, A.*, *Risnack J*, a. *C. Y. Baldwin*: Exploring the Structure of Complex Software Design: An Empricial Study of Open Source and Proprietary Code; in: Management Science, 52 (2006), pp. 1015–1030

*[bibliographic data have be verified]*

*Mahler, Marcus*: Open Source Software: The Success of an Alterntaive Intellectual Property Incentive Paradigm; in: Fordham Intellectual Property, Media & Entertainmeint Law Journal, 21 (2000), pp. 619–646

*[aus widmer] [bibliographic data have be verified]*

*Maldonado, Edgar*: The Process of Introducing FLOSS in the Public Administration: The Case of Venezuela; in: JAIS, 11 (2010), No. 11, pp. 756–783, BibWeb/PDF

*[to be evaluated]*

*Manabe, Yuki*, *Yasuhiro Hayase*, a. *Katuro Inoue*: Evolutional Analysis of Licenses in FOSS; in: Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE); New York, NY, USA: ACM, 2010 (= IWPSE-EVOL '10) ⟨URL: `http://doi.acm.org/10.1145/1862372.1862391`⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–4503–0128–2, pp. 83–87

*[to be evaluated]*

*Mancinelli, Fabio et al.*: Managing the Complexity of Large Free and Open Source Package-Based Software Distributions; in: Proceedings of the 21st IEEE/ACM International Con-

ference on Automated Software Engineering; Washington, DC, USA: IEEE Computer Society, 2006 ⟨URL: http://dl.acm.org/citation.cfm?id=1169218.1169319⟩, ISBN 0–7695–2579–2, pp. 199–208
*[bibliographic data have be verified]*

*Mann, Florian et al.*: Open Access Publishing In Science; in: Communications of the ACM, 52 March (2009), pp. 135–139 ⟨URL: http://doi.acm.org/10.1145/1467247.1467279⟩
*[to be evaluated]*

*Mannaert, Herwig* a. *Kris Ven*: The Use of Open Source Software Platforms by Independent Software Vendors: Issues and Opportunities; In Proceedings of the Fifth Workshop on Open Source Software Engineering, 2005, pp. 7:1–7:4 ⟨URL: http://doi.acm.org/10.1145/1082983.1083266⟩ – reference download: 2011-12.29, BibWeb/PDF
*[to be evaluated]*

*Marly, Jochen*: Praxishandbuch Softwarerecht; 5th edition. München: Beck, 2009
*next action, got from Schichl, evaluate espacially Partl 4, V. [bibliographic data have be verified]*

*Marmorstein, Robert*: Open Source Contribution As An Effective Software Engineering Class Project; in: Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education; New York, NY, USA: ACM, 2011 (= ITiCSE '11) ⟨URL: http://doi.acm.org/10.1145/1999747.1999823⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0697–3, pp. 268–272
*[to be evaluated]*

*Martins Melo, Felipe* a. *Pereira, Jr.*: A Component-Based Open-Source Framework for General-Purpose Recommender Systems; in: Proceedings of the 14th international ACM Sigsoft symposium on Component based software engineering; New York, NY, USA: ACM, 2011 (= CBSE '11) ⟨URL: http://doi.acm.org/10.1145/2000229.2000239⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–4503–0723–9, pp. 67–72
*[to be evaluated]*

*McAllister, Neil*: Licence to Profit [. Hybrid Open Source Licensing]; in: New Architect and Web Techniques (www.newarchitectmag.com), 8 (2003), p. np., Copy
*A very short articel. But it describes how mysql organizes its' dual licensing: they let the external contributors transfer their copyright to MySql AB. By this step the company receives also 'the right to license the (modifed) product under any license (they) wish'.*

*McGowan, D.*: Legal Implications of Open Source Software; in: University Illinois Law Review, (2001), pp. 241–304
*[bibliographic data have be verified]*

*McGowan, David*: The Tory Anarchism of F/OSS Licensing; in: University of Chicago Law Review, 78 (2011), pp. 207–223, BibWeb/PDF
*[to be evaluated]*

*McInerney, Paul-Brian*: Technology Movements and the Politics of Free/Open Source Software; in: Science, Technology & Human Values, 34 (2009), No. 2, pp. 206–233 ⟨URL: http://sth.sagepub.com/content/34/2/206.abstract⟩, BibWeb/PDF
*[to be evaluated]*

*Megias, David et al.*: Free Technology Academy: a European initiative for distance education about Free Software and Open Standards; in: Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education; New York, NY, USA: ACM, 2009 (= ITiCSE '09) ⟨URL: http://doi.acm.org/10.1145/1562877.1562904⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–60558–381–5, pp. 70–74
*[to be evaluated]*

*Meretz, Stefan*: Linux & Co : freie Software - Ideen für eine andere Gesellschaft; Neu-Ulm: ???, 2000
*Präsenz. FaM. Bibliothek Informatik. D.4-368 [bibliographic data have be verified]*

*Metzger, Axel*: Frei ab 18 Jahre; in: Linux-Magazin, (2000), No. 11, pp. 52ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Metzger, Axel*: Anmerkungen zum Urteil vom 19.5.2004 des LG München I zur Wirksamkeit einer GPL-Lizenz; in: CR [Computer und Recht], (2004), pp. 778–780
*[aus oberhem] [bibliographic data have be verified]*

*Michaelson, Jay*: There's no such thing as a Free (Software) Lunch; in: Queue, 2 May (2004), pp. 40–47 ⟨URL: `http://doi.acm.org/10.1145/1005062.1005066`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Microsoft*: Einige Fragen zur GNU General Public License (GPL), die sich jedes Unternehmen stellen sollte; in: ??? (2001)
*aus Koeglin2007a [bibliographic data have be verified]*

*Mitre*: Use of Free and Open Source Software (FOSS) in the U.S. Department of Defense; ⟨URL: `http://www.egovos.org/pdf/dodfoss.pdf`⟩
*[next action, aus Widmer, vergleicht begriffe open source und free software] [bibliographic data have be verified]*

*MLA*: MLA Handbook for Writers of Research Papers; 7th edition. New York: The Modern Language Association of America, 2009, Print, ISBN 978–1–60329–024–1
*The American writer style, whose merits we know and whom we nevertheless do not follow.*

*Mockus, Audris*, *Roy T. Fielding*, a. *James Herbsleb*: A Case Study of Open Source Software Development: the Apache Server; in: Proceedings of the 22nd international conference on Software engineering; New York, NY, USA: ACM, 2000 (= ICSE '00) ⟨URL: `http://doi.acm.org/10.1145/337180.337209`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–206–9, pp. 263–272
*[to be evaluated]*

*Mockus, Audris*, *Roy T. Fielding*, a. *James D. Herbsleb*: Two Case Studies of Open Source Software Development: Apache and Mozilla; in: Transactions on Software Engineering Methodology, 11 July (2002), No. 3, pp. 309–346 ⟨URL: `http://doi.acm.org/10.1145/567793.567795`⟩
*[to be evaluated]*

*Moglen, Peter*: Anarchism triumphant: Free Software and the Death of Copyright; in: First Monday, 48 (1999), p. o.A.
*[bibliographic data have be verified]*

*Monden, A. et al.*: Guilty or Not Guilty: Using Clone Metrics to Determine Open Source Licensing Violations; in: Software, IEEE, 28 march-april (2011), No. 2, pp. 42 –47, ISSN 0740–7459
*[bibliographic data have be verified]*

*Montante, Robert*: A Survey of Portable Software; in: JCSC, 24 January (2009), No. 3, pp. 19–24 ⟨URL: `http://dl.acm.org/citation.cfm?id=1409873.1409879`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Moody, Glyn*: Die Software-Rebellen. Die Erfolgsstory von Linus Torvalds und Linux; transl. from the American [edition, 2000] by Annemarie Pumpering; Landsberg am Lech: verlag moderne industrie, 2001, Print, ISBN 3–478–38730–2
*If your friend only wants to read one book on the topic 'Open Source', then give him this! It tells the story - of nearly all aspects.*

*Moody, Glyn*: Rebel Code: Linux And The Open Source Revolution; [New York]: Basic Books, 2002, Print, ISBN ISBN 978–0738206707

*The original, last version: If your friend only wants to read one book on the topic 'Open Source', then give him this! It tells the story - of nearly all aspects.*

*Moody, Glyn*: Interview with Eric Raymond; in: Linux Journal, 165 January (2008), p. 5:1 ⟨URL: `http://dl.acm.org/citation.cfm?id=1344189.1344194`⟩ – reference download: 2011-12-29, BibWeb/HTML

*[to be evaluated]*

*Morasca, Sandro*, *Davide Taibi*, a. *Davide Tosi*: Towards Certifying the Testing Process of Open-Source Software: New Challenges or Old Methodologies? In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009, pp. 25–30 ⟨URL: `http://dx.doi.org/10.1109/FLOSS.2009.5071356`⟩ – reference download: 2011-12-29, BibWeb/PDF

*[to be evaluated]*

*Morasca, Sandro*, *Davide Taibi*, a. *Davide Tosi*: Towards certifying the testing process of Open-Source Software: New challenges or old methodologies? in: Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development; Washington, DC, USA: IEEE Computer Society, 2009 (= FLOSS '09) ⟨URL: `http://dx.doi.org/10.1109/FLOSS.2009.5071356`⟩, ISBN 978–1–4244–3720–7, pp. 25–30

*[bibliographic data have be verified]*

*Morelli, Ralph* a. *Trishan de Lanerolle*: Foss 101: Engaging Introductory Students in the Open Source Movement; in: Proceedings of the 40th ACM technical symposium on Computer science education; New York, NY, USA: ACM, 2009 (= SIGCSE '09) ⟨URL: `http://doi.acm.org/10.1145/1508865.1508977`⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–60558–183–5, pp. 311–315

*[to be evaluated]*

*Morgan, Lorraine* a. *Patrick Finnegan*: Open Innovation in Secondary Software Firms: An Exploration of Managers' Perceptions of open Source Software; in: SIGMIS Database, 41 February (2010), No. 1, pp. 76–95 ⟨URL: `http://doi.acm.org/10.1145/1719051.1719056`⟩, BibWeb/PDF

*[to be evaluated]*

*Mozilla Foundation*: Mozilla Public License 2.0 (MPL-2.0); 2012, FreeWeb/HTML ⟨URL: `http://www.mozilla.org/MPL/2.0/`⟩ – reference download: 2013-03-05

*The Mozilla Public License 2.0 as it is offered by the Mozilla Foundation*

*Mozilla Foundation*: About MPL 2.0: Revision Process and Changes FAQ; 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://www.mozilla.org/MPL/1.1/`⟩ – reference download: 2013-03-05

*Differences between the MPL-1.1 and MPL-2.0: The mozilla foundation itself says thatt the most important part of the license - the file-level copyleft - is essentially the same in MPL 2.0 and MPL 1.1.*

*Mozilla Foundation*: Mozilla Public License Version 1.1; 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://www.mozilla.org/MPL/1.1/`⟩ – reference download: 2013-03-05

*The Mozilla Public License 1.1 as it is offered by the Mozilla Foundation*

*Mtsweni, Jabu* a. *Elmarie Biermann*: An investigation into the implementation of open source software within the SA government: an emerging expansion model; in: Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology; New York, NY, USA: ACM, 2008 (= SAICSIT '08) ⟨URL: `http://doi.acm.org/10.1145/1456659.1456677`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–60558–286–3, pp. 148–158

*[to be evaluated]*

*Müller Molina, Arnoldo José* a. *Takeshi Shinohara*: On Approximate Matching of Programs for Protecting Libre Software; in: Proceedings of the 2006 Conference of the Center for Advanced Studies on Collaborative research; New York, NY, USA: ACM, 2006 (= CASCON '06) ⟨URL: `http://doi.acm.org/10.1145/1188966.1188994`⟩ – reference download: 2011-12-29, BibWeb/PDF, pp. 1–14

*[to be evaluated]*

*Mundhenke, Jens*: Wettbewerbswirkungen von Open-Source-Software und offenen Standards auf Softwaremärkten; Berlin, Heidelberg, and New York: Springer, 2007 (= Kiel Studies, [Vol./No.] 338), Print, ISBN 978–540–71415–6

*This book asks why and how Open Source Software successfully works as part of the economic markets. One answer shall be that OSS increments the competitive pressure by offering an alternative. Fort analyzing the topic the book follows the 'natural' structure of handling Open Source: it explains the concept, classifies the licenses as Copyleft, Noncopyleft, Public Domain and proprietary, and gives a summary of the OSS history.*

*Munga, Neeshal*, *Thomas Fogwill*, a. *Quentin Williams*: The Adoption of Open Source Software in Business Models: A Red Hat and IBM Case Study; in: Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists; New York, NY, USA: ACM, 2009 (= SAICSIT '09) ⟨URL: `http://doi.acm.org/10.1145/1632149.1632165`⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–60558–643–4, pp. 112–121

*[to be evaluated]*

*Mustaquim, Moyen Mohammad*: A Systems Thinking Model for Open Source Software Development in Social Media; in: Proceedings of the International Workshop on Modeling Social Media; New York, NY, USA: ACM, 2010 (= MSM '10) ⟨URL: `http://doi.acm.org/10.1145/1835980.1835987`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0229–6, pp. 7:1–7:2

*[to be evaluated]*

*Mustonen, Mikko*: Copyleft - the economics of Linux and other open source software; in: Information Economics and Policy, 15 (2003), No. 1, pp. 99–121 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S0167624502000902`⟩ – reference download: 2012-02-09, BibWeb/PDF & Copy

*[to be evaluated]*

*Mustonen, Mikko*: Essays on the Economics of Information and Communication Technologies: Copyleft, Networks and Compatibility; Ph. D thesis, University of Helsinki, Department of Economics, Faculty of Social Sciences, 2003

*next action, aus buchtala [bibliographic data have be verified]*

*Mustonen, Mikko*: Why do firms support the development of substitute copyleft programs? Volume 15 of *Mustonen*: Copyleft - the economics of Linux and other open source software, 2003⟨URL: `http://www.sciencedirect.com/science/article/pii/S0167624502000902`⟩ – reference download: 2012-02-09, pp. ??–??, BibWeb/PDF & Copy

*next action, aus buchtala [bibliographic data have be verified]*

*Mustonen, Mikko*: When Does a Firm Support Substitute Open Source Programming? in: Journal of Economics & Management Strategy, 14 (2005), No. 1, pp. 121–138

*next action, aus buchtala [bibliographic data have be verified]*

*Müller, Martin*: Open Source - kurz & gut; Köln, 1999 ⟨URL: `http://www.oreilly.de/german/freebooks/os_tb/toc.html`⟩

*[aus widmer] [bibliographic data have be verified]*

*Müller-Seitz, Gordon* a. *Guido Reger*: Is open source software living up to its promises? Insights for open innovation management from two open source software-inspired projects; in: R&D Management, 39 (2009), No. 4, pp. 372–381 ⟨URL: http://dx.doi.org/10.1111/j.1467-9310.2009.00565.x⟩ – reference download: 2012-02-09, BibWeb/PDF
*[to be evaluated]*

*Nadah, Nadia*, *Mélanie Dulong de Rosnay*, a. *Bruno Bachimont*: Licensing Digital Content With A Generic Ontology: Escaping From The Jungle of Rights Expression Languages; in: Proceedings of the 11th International Conference on Artificial Intelligence and Law; New York, NY, USA: ACM, 2007 (= ICAIL '07) ⟨URL: http://doi.acm.org/10.1145/1276318.1276330⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–59593–680–6, pp. 65–69
*[to be evaluated]*

*Nagy, Del*, *Areej M. Yassin*, a. *Anol Bhattacherjee*: Organizational adoption of open source software: barriers and remedies; in: Communications of the ACM, 53 (2010), pp. 148–151 ⟨URL: http://doi.acm.org/10.1145/1666420.1666457⟩, BibWeb/PDF
*[to be evaluated]*

*Nakakoji, Kumiyo et al.*: Evolution Patterns of Open-Source Software Systems and Communities; in: Proceedings of the International Workshop on Principles of Software Evolution; New York, NY, USA: ACM, 2002 (= IWPSE '02) ⟨URL: http://doi.acm.org/10.1145/512035.512055⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–545–9, pp. 76–85
*[to be evaluated]*

*Netcraft*: August 2011 Web Server Survey; 2011, FreeWeb/Html ⟨URL: http://news.netcraft.com/archives/2011/08/05/august-2011-web-server-survey-3.html⟩ – reference download: 2011-08-31
*Monthly offered statistic counting sites and their delivering web servers: Following this sheet the apache http server is the most often used software since 1996. And currently it is used more than twice as much as all the others together.*

*Neumann, Peter G.*: Inside Risks: Robust Open-Source Software; in: Communications of the ACM, 42 February (1999), No. 2, p. 128 ⟨URL: http://doi.acm.org/10.1145/293411.293491⟩, BibWeb/PDF
*[to be evaluated]*

*Nilendu, P.* a. *T. R. Madanmohan*: Competing on Open Source: Strategies and Practise; 2002 ⟨URL: http://opensource.mit.edu/papers/madanmohan.pdf⟩
*[bibliographic data have be verified]*

*Noll, John* a. *Wei-Ming Liu*: Requirements Elicitation in Open Source Software Development: A Case Study; [General Chairs: Justin Erenkrantz and Hyrum K. Wright]; In Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2010, pp. 35–40 ⟨URL: http://doi.acm.org/10.1145/1833272.1833279⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Nordquist, Pete*, *Anna Petersen*, a. *Angelina Todorova*: License Tracing in Free, Open, and Proprietary Software; in: JCSC, 19 December (2003), No. 2, pp. 101–112 ⟨URL: http://dl.acm.org/citation.cfm?id=948785.948802⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Nov, Oded* a. *George Kuk*: Open source content contributors' response to free-riding: The effect of personality and context; in: Computers in Human Behavior, 24 (2008), pp. 2848–2861, BibWeb/PDF
*[tbd: evaluate copy]*

*Oberhem, Carolina*: Vertrags- und Haftungsfragen beim Vertrieb von Open Source Software; Dissertation; Hamburg: Verlag Dr. Kovač, 2008 (= Recht der Neuen Medien, [Vol./No.] 50), Print, ISBN 978–3–8300–4075–0

*The book analyzes the liability of authors and distributors of GPL licensed software. This is necessary because in Germany the NO-Warranty clauses of the GPL are not applicable. They are substituted by the rules of 'gifts'. In this sense private authors and distributors are only liable in case of acting deliberately ["vorsätzlich"] or strongly negligently ["grob fahrlässig"]. Additionally commercial distributors must thouroughly check whether a program might damage a customer. But commercial distributors have the full liability for their additionally offered specific services. Beside the 'AGB' character and the typus of GPL as a contract the book refers the idea of Open Source seriously.*

*Oezbek, Christopher*, *Lutz Prechelt*, a. *Florian Thiel*: The Onion has Cancer: Some Social Network Analysis Visualizations of Open Source Project Communication; [General Chairs: Justin Erenkrantz and Hyrum K. Wright]; In Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2010, pp. 5–10 ⟨URL: `http://doi.acm.org/10.1145/1833272.1833274`⟩ – reference download: 2012-02-01, BibWeb/PDF

*[to be evaluated]*

*O'Hara, Keith J.* a. *Jennifer S. Kay*: Open source software and computer science education; in: J. Comput. Small Coll. 18 February (2003), pp. 1–7 ⟨URL: `http://dl.acm.org/citation.cfm?id=771712.771716`⟩, ISSN 1937–4771

*[bibliographic data have be verified]*

*Omsels, Hermann-Josef*: Open Source und das deutsche Vertrags- und Urheberrecht; in: *Christian Schertz* a. *Herman-Josef Omsels, editors:* Festschrift für Paul W. Hertin zum 60. Geburtstag; 2000

*aus Koeglin2007a [bibliographic data have be verified]*

*Open Source Development Labs*: OSDL announces patent common project; 2005 ⟨URL: `http://www.osdl.org/newsroo,/press_releases/2005/2005_08_09_beaverton.html`⟩

*[aus Asche u. Bauhaus] [bibliographic data have be verified]*

*Open Source Initiative*: GNU General Public License, version 2 (GPL-2.0). Version 2, June 1991; 1991 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/GPL-2.0`⟩ – reference download: 2013-02-05

*The Gnu Public License 2.0 as it is offered by the OSI*

*Open Source Initiative*: The GNU Lesser General Public License, version 2.1 (LGPL-2.1); 1999 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/LGPL-2.1`⟩ – reference download: 2013-03-06

*The Lesser Gnu Public License 2.1, as it is offered by the OSI*

*Open Source Initiative*: Common Development and Distribution License (CDDL-1.0); 2004 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/CDDL-1.0`⟩ – reference download: 2013-04-19

*The Common Development and Distribution License 1.0 as it is offered by the OSI*

*Open Source Initiative*: Apache License, Version 2.0; 2004 [n.y. of the page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/Apache-2.0`⟩ – reference download: 2013-02-07

*The Apache 2.0 license, as it is presented by the OSI*

*Open Source Initiative*: Eclipse Public License, Version 1.0; 2005 [n.y. of the page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/EPL-1.0`⟩ – reference download: 2013-02-20

*The Eclipse Public License 1.0, as it is presented by the OSI*

*Open Source Initiative*: European Union Public License, version 1.1 (EUPL-1.1; 2007 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/EUPL-1.1`⟩ – reference download: 2013-03-04

*In contrast to most of the other approved open source licenses, the OSI is offering the European Union Public Licences only as a page linked to the multi lingual versions of EUPL.*

*Open Source Initiative*: GNU Affero General Public License, Version 3 (AGPL-3.0); 2007 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/AGPL-3.0`⟩ – reference download: 2013-04-05

*The Affero Gnu Public License 3.0 as it is offered by the OSI*

*Open Source Initiative*: GNU General Public License, version 3 (GPL-3.0); 2007 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/GPL-3.0`⟩ – reference download: 2013-03-05

*The Gnu Public License 3.0 as it is offered by the OSI*

*Open Source Initiative*: The GNU Lesser General Public License, version 3.0 (LGPL-3.0); 2007 [n.y. of the html page itself], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/LGPL-3.0`⟩ – reference download: 2013-03-06

*The Lesser Gnu Public License 3.0, as it is offered by the OSI*

*Open Source Initiative*: The BSD 2-Clause License; 2012 [n.y.], FreeWeb/HTML ⟨URL: `http://www.opensource.org/licenses/BSD-2-Clause`⟩ – reference download: 2012-07-03

*The latest current BSD license, known as 2 clause BSD license.*

*Open Source Initiative*: The BSD 3-Clause License; 2012 [n.y.], FreeWeb/HTML ⟨URL: `http://www.opensource.org/licenses/BSD-3-Clause`⟩ – reference download: 2012-07-04

*The elder BSD license, known as 3 clause BSD license.*

*Open Source Initiative*: The MIT License; 2012 [n.y.], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/mit-license.php`⟩ – reference download: 2012-08-24

*The MIT license, the most permissive Open Source License*

*Open Source Initiative*: The Open Source Definition; 2012 [n.y.], FreeWeb ⟨URL: `http://www.opensource.org/docs/osd`⟩ – reference download: 2012-06-21

*This page lists the ten rules which together specify under which conditions a specific license can be classified as Open Source license. It presents the Open Source Definition. So, a piece of software is only then Open Source Software, if its license has been approved as an Open Source License by the Open Source Initiative.*

*Open Source Initiative*: The Open Source Initiative; 2012 [n.y.], FreeWeb ⟨URL: `http://www.opensource.org/about/`⟩ – reference download: 2013-01-22

*By this page the Open Source Initiative describes itself, its' targets and goals.*

*Open Source Initiative*: The Open Source Licenses, alphabetically sorted; 2012 [n.y.], FreeWeb ⟨URL: `http://opensource.org/licenses/alphabetical`⟩ – reference download: 2013-01-22

*This page lists the approved Open Source Licenses*

*Open Source Initiative*: The [OSI] Licence Review Process; 2012 [n.y.], FreeWeb ⟨URL: `http://www.opensource.org/approval`⟩ – reference download: 2013-01-22

*By this page, the OSI describes its' approval process*

*Open Source Initiative*: OSI Mailing List. License-discuss. Draft of new OSI licenses landing page; 2012 [n.y.], FreeWeb/HTML ⟨URL: `http://projects.opensource.org/pipermail/license-discuss/2012-April/000332.html`⟩ – reference download: 2013-01-29

*Thread discussing a new license taxonomy.*

*Open Source Initiative*: Microsoft Public License (MS-PL); 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/MS-PL`⟩ – reference download: 2013-02-26

*The Microsoft Public License, as it is presented by the OSI*

*Open Source Initiative*: Mozilla Public License 2.0 (MPL-2.0); 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/MPL-2.0`⟩ – reference download: 2013-02-07
*The Mozilla Public License 2.0 as it is offered by the OSI*

*Open Source Initiative*: Open Source Licenses by Category; 2013 [n.y.], FreeWeb ⟨URL: `http://opensource.org/licenses/category`⟩ – reference download: 2013-01-29
*This page categorizes the approved Open Source Licenses*

*Open Source Initiative*: The PHP License 3.0 (PHP-3.0); 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/PHP-3.0`⟩ – reference download: 2013-02-27
*The PHP-3.0 license, as it is presented by the OSI*

*Open Source Initiative*: The PostgreSQL Licence (PostgreSQL); 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://opensource.org/licenses/PostgreSQL`⟩ – reference download: 2013-02-27
*The PostgreSQL license, as it is presented by the OSI*

*Oreg, Shaul* a. *Oded Nov*: Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values; in: Computers in Human Behavior, 24 (2008), No. 5, pp. 2055–2073 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S0747563207001537`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*O'Reilly, Tim*: Lessons from Open-Source Software Development; in: Communications of the ACM, 42 (1999), No. 4, pp. 32–37 ⟨URL: `http://doi.acm.org/10.1145/299157.299164`⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*O'Reilly, Tim, editor*: Open Source: kurz und gut; 1999
*aus Koeglin2007a [bibliographic data have be verified]*

*Orsila, Heikki et al.*: Trust Issues in Open Source Software Development; in: Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010; New York, NY, USA: ACM, 2009 (= WUP '09) ⟨URL: `http://doi.acm.org/10.1145/1527033.1527037`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–565–9, pp. 9–12
*[to be evaluated]*

*Osterloh, M., S. Rota*, a. *M. von Wartburg*: Open Source - New Rules in Software Developement; Zürich, 2001, working paper ⟨URL: `http://www.iou.unizh,ch/orga/downloads/OpenSourceAoM.pdf`⟩
*[bibliographic data have be verified]*

*Osterloh, Margit* a. *Sandra Rota*: - Just another case of collective invention? in: Research Policy, 36 (2007), No. 2, pp. 157–171 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S0048733306001983`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Osterloh, Margit*, *Sandra Rota*, a. *Bernhard Kuster*: Open Source Software Production: Climbing on the Shoulders of Giants; 2002 ⟨URL: `http://opensource.mit.edu/papers/osterlohrotakuster.pdf`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Osterloh, Margit*, *Sandra Rota*, a. *Roger Lüthi*: 'Collective Invention' als neues Innovationsmodell; In *Drossou, Krempl, a. Poltmann*: Die wunderbare Wissensvermehrung, 2006, pp. 65–76, Print
*The article shows that 'collective invention' can already be discovered in the 19th century. 'Collective invention' denotes the free exchange and development of ideas. In the past it took place in the volatile not productive phase of thinking and stopped when the main architecture / design had been invented. OS development doesn't follow this pattern: its' free exchange is preserved because of the low costs of participation and the OS licenses which demand the 'Giving-Back' of inventions.*

*O'Sullivan, Maureen*: Eof[:] Free Software Licenses; in: Linux Journal, 122 June (2004), pp. Article No. 11 ⟨URL: http://dl.acm.org/citation.cfm?id=993247.993258⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*O'Mahony, Siobhán*: Guarding the commons: how community managed software projects protect their work; in: RP, 32 (2003), No. 7, pp. 1179–1198 ⟨URL: http://www.sciencedirect.com/science/article/pii/S0048733303000489⟩ – reference download: 2012-02-09, BibWeb/PDF
*[to be evaluated]*

*Patterson, Chip*: Copyright Misuse and Modified Copyleft: New Solutions to the Challenges of Internet; in: Michigan Law Review, 98 (2000), No. 5, pp. 1351–1383, BibWeb/PDF
*[to be evaluated]*

*Pelizza, Annalisa*: Openness as an Asset: A Classification System for Online Communities Based on Actor-Network Theory; in: Proceedings of the 6th International Symposium on Wikis and Open Collaboration; New York, NY, USA: ACM, 2010 (= WikiSym '10) ⟨URL: http://doi.acm.org/10.1145/1832772.1832784⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0056–8, pp. 8:1–8:10
*[to be evaluated]*

*Perens, Bruce*: The Open Source Definition; In *DiBona, Ockman, a. Stone*: Open Sources, 1999, pp. 171–188
*next action, aus buchtala [bibliographic data have be verified]*

*Perens, Bruce*: Combining GPL and Proprietary Software; in: Datamation, 9 (2009), pp. wp. [3 pages] ⟨URL: http://www.datamation.com/osrc/article.php/3801396/Bruce-Perens-Combining-GPL-and-Proprietary-Software.htm⟩ – reference download: 2012-03-09, FreeWeb/HTML
*dowload*

*Perr, Jon*, *Melissa M. Appleyard*, a. *Patrick Sullivan*: Open for business: emerging business models in open source software; in: INTERNATIONAL JOURNAL OF TECHNOLOGY MANAGEMENT, 52 (2010), pp. 432–456
*next action, found in ISI Web of Knowledge [bibliographic data have be verified]*

*Peters, Stormy*: Open Source Is Changing the Way Work Gets Done; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, p. 1, BibWeb/PDF
*Keynote. Affirms that the new methods of collaboration and communication, being used by Open Source development, will also change the cooparation of individuals and companies.*

*Petreley, Nicholas*: /var/opinion: The GPLv2 vs. GPLv3 Debate; in: Linux Journal, 153 January (2007), p. 17 ⟨URL: http://dl.acm.org/citation.cfm?id=1194955.1194972⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Phillips, Douglas E.*: The Software License Unveiled. How Legislation by License Controls Software Access; Oxford, New York, Auckland [etc. ...]: Oxford University Press, 2009, ISBN 978–0–19–534187–4
*This book contrasts the proprietary licenses and their Free and/or Open Source alternatives. The conclusion is daring: Proprietary software is described as being focused on the usability (and disregarding the rights of the user), Open Source Software as being focused on the developer (and neglecting the usability).*

*Piller, Frank T.*: User Innovation: der Kunde kann's besser; In *Drossou, Krempl, a. Poltmann*: Die wunderbare Wissensvermehrung, 2006, pp. 85–97, Print
*The article describes that the development of new sails for kite-surfing is supported by the end users: Similar to the OS development style they get CAD models freely, let their sails be sewed, and give their improvings as a set of bug reports back to the developer.*

*Piller, Harald*: Von Open Source zu Open Innovation; in: Harvard Business Manager, 25 (2003), No. 12, p. 114
*[bibliographic data have be verified]*

*Pisano, G.*: Profiting from Innovation and the Intellectual Property Revolution; in: Research Policy, 35 (2006), No. 8, pp. 1122–1130
*[bibliographic data have be verified]*

*Plaß, Gunda*: Open Contents im deutschen Urheberrecht; in: GRUR, (2002), pp. 670ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Polanski, Arnold*: Is the General Public Licence a Rational Choice? in: Journal of Industrial Economics, 55 (2007), pp. 691–714, BibWeb/PDF
*[tbd: evaluate copy]*

*Prechelt, Lutz*: Some Non-Usage Data for a Distributed Editor: the Saros Outreach; in: Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering; New York, NY, USA: ACM, 2011 (= CHASE '11) ⟨URL: `http://doi.acm.org/10.1145/1984642.1984651`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–4503–0576–1, p. 48
*[to be evaluated]*

Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development; (= FLOSS '09) Washington, DC, USA: IEEE Computer Society, 2009 ⟨URL: `http://dl.acm.org/citation.cfm?id=1572192`⟩ – reference download: 2012-01-25, BibWeb/PDF, ISBN 978–1–4244–3720–7
*[to be evaluated]*

Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development; [General Chairs: Justin Erenkrantz and Hyrum K. Wright]; (= FLOSS '10) New York, NY, USA: ACM, 2010 ⟨URL: `http://dl.acm.org/citation.cfm?id=1833272`⟩ – reference download: 2012-01-25, BibWeb/PDF, ISBN 978–1–60558–978–7
*[to be evaluated]*

Proceedings of the Fifth Workshop on Open Source Software Engineering; (= 5-WOSSE) New York, NY, USA: ACM, 2005 ⟨URL: `http://doi.acm.org/10.1145/1082983.1083260`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–59593–127–9
*Items seperately listed*

*Qureshi, Israr* a. *Yulin Fang*: Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach; in: Organizational Research Methods, 14 (2011), No. 1, pp. 208–238, BibWeb/PDF
*[tbd: evaluate copy]*

*Rafiq, Muhammad*: LIS community's perceptions towards open source software adoption in libraries; in: International Information & Library Review (2009) 41, 137e145, 41 (2009), pp. 137–145, BIbWeb/PDF
*[to be evaluated]*

*Raja, Uzma* a. *Evelyn Barry*: Investigating quality in large-scale Open Source Software; In Proceedings of the Fifth Workshop on Open Source Software Engineering, 2005, pp. 1–4 ⟨URL: `http://doi.acm.org/10.1145/1082983.1083268`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Raymond, Eric*: A Brief History of Hackerdom, revised version; 2000 ⟨URL: `http://www.catb.org/~esr/writings/hacker-history/hacker-history.html`⟩
*[bibliographic data have be verified]*

*Raymond, Eric S.*: How To Become A Hacker; ⟨URL: `http://www.catb.org/esr/faqs/hacker-howto.html`⟩

*[next action aus Widmer] [bibliographic data have be verified]*

*Raymond, Eric S.*: The Cathedral and the Bazaar; in: First Monday, 3 March (1998), No. 3, p. o.A.
*next action, aus buchtala [bibliographic data have be verified]*

*Raymond, Eric S.*: Homesteading the Noosphere: An Introductory Contradiction; in: First Monday, 3 (1999), No. 10, p. o.A.
*next action, aus buchtala [bibliographic data have be verified]*

*Raymond, Eric S.*: The cathedral and the bazaar : musings on Linux and open source by an accidental revolutionary; Peking [...]: ???, 2001
*Präsenz. FaM. UB. Bibliothekszentrum Geisteswissenschaften. IG Farben-Haus, Q1, 5. OG, Raum 5.111. 14/X. a. 27 [bibliographic data have be verified]*

*Reed, Matthew W. et al.*: Developing and Learning Web Services with Open Source Software: An Experience Report; in: JCSC, 22 April (2007), No. 4, pp. 93–100 ⟨URL: `http://dl.acm.org/citation.cfm?id=1229637.1229654`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Reese, Björn* a. *Daniel Sternberg*: Working without Copyleft; 2001 ⟨URL: `http://www.oreillynet.com/lpt/a/1403`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Reincke, Karsten*: Classical Scholar Texts With Footnotes based on LaTeX, BibTeX, Koma, jurabib and mykeds-CSR; 2012, FreeWeb/Html ⟨URL: `http://www.fodina.de/en/closedprojects/latex-addons/classical-scholar.html`⟩ – reference download: 2013-02-10
*Short description and demonstration of a subtler use of secondary literature*

*Reincke, Karsten*: (Geistes-) Wissenschaftliche Texte mit jurabib. Dienst am Leser, Dienst am Scholaren: Uber Anmerkungsapparate in Fußnoten - aber richtig. [n.l.], 2012 ⟨URL: `http://download.fodina.de/fodinaClassicalScholarFoNoDe.pdf`⟩ – reference download: 2013-02-10, FreeWeb/PDF
*More legitimizing details on a subtler method to use secondary literature adequately.*

*Reitmayr, Gerhard* a. *Dieter Schmalstieg*: An Open Software Architecture for Virtual Reality Interaction; in: Proceedings of the ACM Symposium on Virtual Reality Software and Technology; New York, NY, USA: ACM, 2001 (= VRST '01) ⟨URL: `http://doi.acm.org/10.1145/505008.505018`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–427–4, pp. 47–54
*[to be evaluated]*

*Renner, Thomas et al.*: Open Source Software. Einsatzpotentiale und Wirtschaftlichkeit; eine Studien der Fraunhofer Gesellschaft; Stuttgart: Fraunhofer IRB Verlag, 2005, Print, ISBN 3–8167–7008–8
*The study discusses potentials of OSS and offers a method to compute its' cost effectiveness. It presents a good survey of the underlying concepts and a list of OS applications. For client mashines the computation of the cost effectiveness refers only to the migration from MS-Office to Open-Office. Unfortunately the list of advantages and disadvantages based on interviews is a little inconsistent. Remarkably is also that the study doesn't focus on the act of fulfilling an OS license although it highlights, that OSS is not free of license conditions.*

*Reynolds, Carl J* a. *Jeremy C Wyatt*: Open Source, Open Standards, and Health Care Information Systems; in: JMIR, 13 (2011), No. 1, p. wp ⟨URL: `http://www.jmir.org/2011/1/e24/`⟩, BibWeb/HTML
*[to be evaluated]*

*Rigby, Peter C.*, *Daniel M. German*, a. *Margaret-Anne Storey*: Open Source Software Peer Review Practices: A Case Study of the Apache Server; in: Proceedings of the 30th International Conference on Software Engineering; New York, NY, USA: ACM, 2008 (= ICSE

'08) ⟨URL: http://doi.acm.org/10.1145/1368088.1368162⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–60558–079–1, pp. 541–550
*[to be evaluated]*

*Rigby, Peter C.* a. *Margaret-Anne Storey*: Understanding Broadcast Based Peer Review on Open Source Software Projects; in: Proceedings of the 33rd International Conference on Software Engineering; New York, NY, USA: ACM, 2011 (= ICSE '11) ⟨URL: http://doi.acm.org/10.1145/1985793.1985867⟩ – reference download: 2011-12-29, BibWen/PDF, ISBN 978–1–4503–0445–0, pp. 541–550
*[to be evaluated]*

*Rivlin, Gary*: Linus Torvalds - Leader of the Free World; in: Wired Magazin, (2003), No. 11, pp. 152ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Robbins, Arnold*: What's GNU? in: Linux Journal, 1 March (1994), p. 9:1 ⟨URL: http://dl.acm.org/citation.cfm?id=328204.328213⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Robert W, Guomulkiewics*: How Copyleft uses License Right to succeed in the OPen Source Software Revolution and the Implications fpr Article 2b; in: Houston Law Review, 36 (???), pp. 179ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Roberts, Keith A.*: Generic Methodology for Open Source Software Development; in: SIGSOFT Software Engineering Notes, 30 March (2005), No. 2, pp. 1–5 ⟨URL: http://doi.acm.org/10.1145/1050849.1050863⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Rose, Marshall T.*: The Open Book, A Practical Perspective on OSI; Englewood Cliffs NJ: Prentice Hall, 1990, Print, ISBN 0–13–643016–3
*In the most cases 'OSI' refers the 'Open Systems Interconnection Model', not the 'Open Source Initiative'. Here an English written book explaining the elder meaning of OSI as a model of network layers.*

*Rosen, Lawrence*: Geek Law[:] A Question of Licenses; in: Linux Journal, 89 September (2001), p. 14 ⟨URL: http://dl.acm.org/citation.cfm?id=509824.509838⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Copyright Questions; in: Linux Journal, 88 August (2001), p. 13 ⟨URL: http://dl.acm.org/citation.cfm?id=509800.509813⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] License FUD; in: Linux Journal, 92 December (2001), p. 14 ⟨URL: http://dl.acm.org/citation.cfm?id=512620.512634⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Naming Open-Source Software; in: Linux Journal, 90 October (2001), p. 11 ⟨URL: http://dl.acm.org/citation.cfm?id=509852.509863⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Allocation of the Risks; in: Linux Jo, 101 September (2002), p. 17 ⟨URL: http://dl.acm.org/citation.cfm?id=566949.566966⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Bad Law; in: Linux Journal, 98 June (2002), p. 13 ⟨URL: http://dl.acm.org/citation.cfm?id=513489.513502⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Dealing with Patents in Softwar Licences; in: Linux Journal, 93 January (2002), p. 14 ⟨URL: http://dl.acm.org/citation.cfm?id=512788.512802⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Dealing With Patents in Software Licenses, Part II; in: Linux Journal, 94 February (2002), p. 15 ⟨URL: http://dl.acm.org/citation.cfm?id=513039.513054⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Fair Use; in: Linux Journal, 100 August (2002), p. 18 ⟨URL: http://dl.acm.org/citation.cfm?id=563953.563971⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] License Defamation; in: Linux Journal, 99 July (2002), p. 15 ⟨URL: http://dl.acm.org/citation.cfm?id=513581.513596⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Unbiased License FUD; in: Linux Journal, 95 March (2002), p. 15 ⟨URL: http://dl.acm.org/citation.cfm?id=513085.513100⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Geek Law[:] Why the Public Domain Isn't a License; in: Linux Journal, 102 October (2002), p. 12 ⟨URL: http://dl.acm.org/citation.cfm?id=571785.571797⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: IAAL[:] Derivative Works; in: Linux Journal, 105 January (2003), p. 13 ⟨URL: http://dl.acm.org/citation.cfm?id=603771.603784⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Rosen, Lawrence*: Open Source Licensing. Software Freedom and Intellectual Property Law; Upper Saddle River, New Jersey: Prentice Hall PTr, 2005, ISBN 0–13–148787–6
*Well-conceived book which discusses different aspects by contrasting the Academic Licenses (also known as permissive Open Source Licenses) and the Reciprocal Licenses (also known as copylefted Open Source Licenses). But one has to be careful: to define the opposite of ASL and OSL by the opposite of recirpocal and not reciprocal licenses does not match the set of OSL in the sense of the OSI.*

*Rosen, Lawrence*: OSL 3.0: A Better License for Open Source Software; in: CRi, 6 (2007), pp. 166–171, Copy
*FernleihCopy*

*Rosenberg, D. K.*: Open source - The unauthorized white papers; Chicago, 2000
*[aus Asche u. Bauhus] [bibliographic data have be verified]*

*Rossi, Christina* a. *Andrea Bonaccorsi*: Intrinsic motivations and preit-oriented firms supplying Open Source products and services; in: First Monday, 10 May (2005), No. 5, p. o.a.
*next action, aus buchtala [bibliographic data have be verified]*

*Rossi, Cristina* a. *Andrea Bonaccorsi*: Why profit-oriented companies enter the OS field?: Intrinsic vs. extrinsic incentives; In Proceedings of the Fifth Workshop on Open Source

Software Engineering, 2005, pp. 12:1–12:5 ⟨URL: http://doi.acm.org/10.1145/1082983.1083269⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Rossi, Naria Alessandra*: Decoding the "Fre/Open Sour ce(F/OSS) Software Puzzle": a survey of theoretical and empirical contributions; 2004 ⟨URL: http://opensource.mit.edu/papers/rossi.pdf⟩
*[bibliographic data have be verified]*

*Ruffin, M.* a. *C. Ebert*: Using Open Source Software in Product Development: A primer; in: IEEE SOFTWARE, 21 (2004), No. 1, pp. 82–86
*[bibliographic data have be verified]*

*Sabin, Mihaela*: Free and Open Source Software Development of IT Systems; in: Proceedings of the 2011 Conference on Information Technology Education; New York, NY, USA: ACM, 2011 (= SIGITE '11) ⟨URL: http://doi.acm.org/10.1145/2047594.2047601⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–1017–8, pp. 27–32
*[to be evaluated]*

*Sakurai, Cledson Akio* a. *Moacyr Martucci Junior*: An Open System Architecture for Operation Support System at Telecommunications Service Providers; in: Proceedings of the 1st International Symposium on Information and Communication Technologies; o.O.: Trinity College Dublin, 2003 (= (ISICT '03)) ⟨URL: http://dl.acm.org/citation.cfm?id=963600.963705⟩ – reference download: 2012-02-01, BibWeb/PDF, pp. 524–529
*[to be evaluated]*

*Samoladas, Ioannis et al.*: Open Source Software Development Should Strive for Even Greater Code Maintainability; in: Communications of the ACM, 47 October (2004), No. 10, pp. 83–87 ⟨URL: http://doi.acm.org/10.1145/1022594.1022598⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Samuelson, Pamela*: IBM's Pragmatic Embrace of Open Source; in: Communications of the ACM, 49 (2006), No. 10, pp. 21–25 ⟨URL: http://doi.acm.org/10.1145/1164394.1164412⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Samuelson, Pamela*: Legally Speaking[:] When is a "License" Really a Sale? in: Communications of the ACM, 52 March (2009), No. 3, pp. 27–29 ⟨URL: http://doi.acm.org/10.1145/1467247.1467258⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Sandred, J.*: Managing open source projects; New York, 2001
*[aus Asche u. Bauhus] [bibliographic data have be verified]*

*Santos Jr., Carlos Denner et al.*: Intellectual Property Policy and Attractiveness: A Longitudinal Study of Free and Open Source Software Projects; in: Proceedings of the ACM 2011 conference on Computer supported cooperative work; New York, NY, USA: ACM, 2011 (= CSCW '11) ⟨URL: http://doi.acm.org/10.1145/1958824.1958950⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 978–1–4503–0556–3, pp. 705–708
*[to be evaluated]*

*Sauer, Robert M.*: Why develop open-source software? The role of non-pecuniary benefits, monetary rewards, and open-source licence type; in: Oxford Review of Economic Policy, 23 (2007), No. 4, pp. 605–619, BibWeb/PDF
*[tbd: evaluate copy]*

*Sauerburger, Heinz, editor*: Open Source Software; dpunkt.verlag, 2004
*[aus Oberhem: wichtig: viele Artikel?] [bibliographic data have be verified]*

*Savage, S. S.*: Conquering Open Source Fears; in: Linux Executive Report, (2006) ⟨URL: http://www.ibm.com/linux/⟩

*[bibliographic data have be verified]*

*Scacchi, W.*: Understanding the Requirements for Developing Open Source Software Systems; in: IEEE Procedings Software, 149 (2002), pp. 24–39
*[bibliographic data have be verified]*

*Scacchi, Walt*: OpenEC/B: Electronic Commerce and Free/Open Source Software Development; In Proceedings of the Fifth Workshop on Open Source Software Engineering, 2005, pp. 8:1–8:5 ⟨URL: http://doi.acm.org/10.1145/1082983.1083270⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Scacchi, Walt*: Free/Open Source Software Development: Recent Research Results and Emerging Opportunities; in: The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering: companion papers; New York, NY, USA: ACM, 2007 (= ESEC-FSE companion '07) ⟨URL: http://doi.acm.org/10.1145/1295014.1295019⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–59593–812–1, pp. 459–468
*[to be evaluated]*

*Scacchi, Walt*: The Future of Research in Free/Open Source Software Development; in: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research; New York, NY, USA: ACM, 2010 (= FoSER '10) ⟨URL: http://doi.acm.org/10.1145/1882362.1882427⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0427–6, pp. 315–320
*[to be evaluated]*

*Schiff, Aaron*: The economics of open source software: A survey of the early literature; in: The Review of Network Economics, 1 March (2002), No. 1, pp. 66–74
*next action, aus buchtala [bibliographic data have be verified]*

*Schiffner, Thomas*: Open Source Software - Freie Software im deutschen Urheber- und Vertragsrecht; München, 2002
*aus Koeglin2007a [bibliographic data have be verified]*

*Schlesinger, David*: Working with Open Source: A Practical Guide; in: interactions, 14 November/December (2007), pp. 35–37 ⟨URL: http://doi.acm.org/10.1145/1300655.1300678⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Schmitz, L.*: Linuxworld: Debatte um Pool für Open-Source-Patente; in: Computerwoche, (2005) ⟨URL: http://www.computerwoche.de/index.cfm?pageid=254\&artid=79815⟩
*[aus Asche u. Bauhus] [bibliographic data have be verified]*

*Schneider, Jochen*: Handbuch des EDV-Rechts; 4th edition. Köln: Dr. Otto Schmidt, 2009
*next action, got by Schichl, behandelt OSS aber eher en passant [bibliographic data have be verified]*

*Schricker* a. *Ulrich Loewenheim, editors*: Urheberrecht; Kommentar; 4th edition. München: Beck, 2010
*next Action, got by Schichl [bibliographic data have be verified]*

*Schryen, Guido*: Is open source security a myth? in: Commununications of the ACM, 54 (2011), pp. 130–140 ⟨URL: http://doi.acm.org/10.1145/1941487.1941516⟩, BibWeb/PDF
*[to be evaluated]*

*Schryen, Guido* a. *Rouven Kadura*: Open source vs. closed source software: towards measuring security; in: *Sung Y. Shin* a. *Sascha Ossowski, editors:* Proceedings of the 2009 ACM symposium on Applied Computing; New York, NY, USA: ACM, 2009 (= SAC '09) ⟨URL: http://doi.acm.org/10.1145/1529282.1529731⟩ – reference download: 2012-01-06, BibWeb/PDF, ISBN 978–1–60558–166–8, pp. 2016–2023
*[to be evaluated]*

*Schulz, Carsten*: VSI-Gutachten zu Open-Source-Software. Die scharfe Klinge des Gesetzes? in: Linux-Magazin, (2003), pp. 68ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Schulz, Carsten*: Dezentrale Softwareentwicklungs- und Softwarevermarktungskonzepte. Vertragsstrukturen in Open Source Modellen; Köln, 2005
*aus Koeglin2007a*

*Schäfer, Fabian*: Der virale Effekt. Entwicklungsrisiken im Umfeld von Open Source Software; Karlsruhe: Universitätsverlag Karlsruhe, 2007, BibWeb/PDF, ISBN 978–3–86644–141–5
*[tbd: evaluate copy]*

*Searls, Doc*: Linux for Suits: Linus Takes a Pass on the New GPL Draft; in: Linux Journal, 145 May (2006), p. 15 ⟨URL: `http://dl.acm.org/citation.cfm?id=1134160.1134175`⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Searls, Doc*: Eof[:] Why to Build on Foss in the First Place; in: Linux Journal, 165 January (2008), pp. Article No. 16 ⟨URL: `http://dl.acm.org/citation.cfm?id=1344189.1344205`⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Searls, Doc*: Eof[:] The Power of Definitions; in: Linux Journal, 177 January (2009), pp. Article No. 15 ⟨URL: `http://dl.acm.org/citation.cfm?id=1502508.1502523`⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Sebald, Gerd*: Offene Wissensökonomie. Analysen zur Wissenssoziologie der Free/Open Source Softwareentwicklung; Dissertation; Wiesbaden: VS Verlag für Sozialwissenschaften, 2008, Print a. BibWeb/PDF, ISBN 978–3–531–15705–4
*From the practical viewpoint a withdrawn socialogical book: tries to discuss the conditions for the possibility that an open knowledge economy establishes itself in a captalist environment. OS seems to be reduced to GPL. License questions are discussed on only 3 pages. But at least the book outlines the evolution of the idea 'Free Software'. And it highlights that the forerunner of the GPL - the emacs license - had still required to publish all changings - even the most private improvements - whereas later on the GPL gave up this condition.*

*Seel, Bernd* a. *Miriam Kraft*: Einführung in das Prinzip Open Source; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 9–19, Print
*This article is an introduction into the topic of the other articles of the collection. For that purpose it offers a really lean summary of the history and the ideas of Open Source. Because of the shortage the article seems to mislead sometimes.*

*Seemayer, Walter* a. *Jason Matusow*: Das Microsoft-Shared-Source-Programm aus der Business-Perspektive; In *Lutterbeck a. Bärwolff*: Open Source Jahrbuch, 2005, pp. 185–200 ⟨URL: `http://...mitaufnehmen..`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Sen, Ravi*, *Chandrasekar Subramaniam*, a. *Matthew Nelson*: Determinants of the Choice of Open Source Software License; in: J. Manage. Inf. Syst. 25 December (2008), pp. 207–240 ⟨URL: `http://dl.acm.org/citation.cfm?id=1554453.1554460`⟩, ISSN 0742–1222
*[bibliographic data have be verified]*

*Sen, Ravi*, *Chandrasekar Subramaniam*, a. *Matthew L. Nelson*: Open source software licenses: Strong-copyleft, non-copyleft, or somewhere in between? in: Decision Support Systems, 52 (2011), No. 1, pp. 199–206 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S0167923611001242`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Sester, Peter*: Open-Source-Software: Vertragsrecht, Haftungsrisiken und IPR-Fragen; in: CR [Computer und Recht], (2000), pp. 797ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Sethanandha, Bhuricha Deen*: Improving Open Source Software Patch Contribution Process: Methods and Tools; in: Proceedings of the 33rd International Conference on Software Engineering; New York, NY, USA: ACM, 2011 (= ICSE '11) ⟨URL: `http://doi.acm.org/10.1145/1985793.1986018`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–4503–0445–0, pp. 1134–1135
*[to be evaluated]*

*Shibuya, Bianca* a. *Tetsuo Tamai*: Understanding the Process of Participating in Open Source Communities; In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009, pp. 1–6 ⟨URL: `http://dx.doi.org/10.1109/FLOSS.2009.5071352`⟩ – reference download: 2012-02-01, BibWeb/PDF
*[to be evaluated]*

*Siepmann, Jürgen*: Lizenz- und haftungsrechtliche Fragen bei der kommerziellen Nutzung Freier Software; in: JurPC Web-Dok, (1999), p. 163
*aus Koeglin2007a [bibliographic data have be verified]*

*Siepmann, Jürgen*: Freie Software - Rechtfreier Raum? Rechtssicherheit im Umgang mit Open Source Software; München, 2000
*aus Koeglin2007a [bibliographic data have be verified]*

*Singh, Param Vir*: The Small-World Effect: The Influence of Macro-Level Properties of Developer Collaboration Networks on Open-Source Project Cuccess; in: Transactions on Software Engineering Methodology, 20 (2010), No. 2, pp. 6:1–6:27 ⟨URL: `http://doi.acm.org/10.1145/1824760.1824763`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Siponen, Mikko*: A Justification for Software Rights; in: SIGCAS, 36 September (2006), No. 3, pp. 11–20 ⟨URL: `http://dl.acm.org/citation.cfm?id=1195716.1195718`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Sirkkala, Petri*, *Timo Aaltonen*, a. *Imed Hammouda*: Opening Industrial Software: Planting an Onion; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 57–69, BibWeb/PDF
*[tbd: evaluate copy]*

*Smith, Bradford L.*: The Future of Software: Enabling the Marketplace to Decide; In *Hahn*: Government Policy toward Open Source Software, 2002
*next action, aus buchtala [bibliographic data have be verified]*

*Sojer, Manuel*: Reusing Open Source Code. Value Creation and Value Appropriation. Perspectives on Knowledge Reuse; wuth a Foreword by Univ.-Prof. Dr. Joachim Henkel; Wiesbaden: Gabler, 2011 (= Gabler Research[:] Innovation und Entrepreneurship), BibWEB/PDF, ISBN 978–8349–2668–5
*eval copy*

*Sojer, Manuel* a. *Joachim Henkel*: License Risks from Ad Hoc Reuse of Code from the Internet; in: Communications of the ACM, 54 December (2011), No. 12, pp. 74–81 ⟨URL: `http://doi.acm.org/10.1145/2043174.2043193`⟩ – reference download: 2011-12-28, BibWeb/PDF
*[to be evaluated]*

*Soto, Martin* a. *Marcus Ciolkowski*: The QualOSS Open Source Assessment Model Measuring the Performance of Open Source Communities; in: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement; Washington, DC, USA:

IEEE Computer Society, 2009 (= ESEM '09) ⟨URL: http://dx.doi.org/10.1109/ESEM.2009.5314237⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–4244–4842–5, pp. 498–501
*[to be evaluated]*

*Sowe, Sulayman K., Ioannis Stamelos*, a. *Lefteris Angelis*: Understanding knowledge sharing activities in free/open source software projects: An empirical study; in: Journal of Systems and Software, 81 (2008), No. 3, pp. 431–446 ⟨URL: http://www.sciencedirect.com/science/article/pii/S0164121207000842⟩ – reference download: 2012-02-03, BibWeb/PDF
*[to be evaluated]*

*Spielkamp, Mathias*: Creative Commons[:] Andere Zeiten, andere Lizenzen; In *Djordjevic et al.*: Urheberrecht im Alltag, 2008, pp. 219–221, Print
*Very short disquisition on the usage of Creative Commons Licences*

*Spielkamp, Mathias*: Lessigletters-Remix[:] Die Creative-Commons-Initiative; In *Djordjevic et al.*: Urheberrecht im Alltag, 2008, pp. 223–230, Print
*Describes the history and the targets of the Creative Commons Licence movement: the principle of reciprocity - for software established by the FSF/GPL - is transferred to the world of texts.*

*Spindler, Gerald*: Rechtsfragen der Open Source Software. Gutachten im Auftrags des VSI; München, 2003
*aus Koeglin2007a [bibliographic data have be verified]*

*Spindler, Gerald*: Stellungnahme [zum Gutachten der VSI]; in: Linux-Magazin, (2003), No. 9, p. 70
*aus Koeglin2007a [bibliographic data have be verified]*

*Spindler, Gerald*: Ausgewählte urheberrechtliche Problem von Open Source Software unter der GPL; in: *Alfread Büllesbacg* a. *Thomas Dreier, editors:* Wem gehört die Information im 21. Jahrhundert; 2004
*aus Koeglin2007a [bibliographic data have be verified]*

*Spindler, Gerald*: Open Source Software auf dem gerichtlichen Prüfstand - Dingliche Qualifikation und Inhaltskontrolle; in: Kommunikation und Recht, (2004), pp. 528–524
*[bibliographic data have be verified]*

*Spindler, Gerald; Spindler, Gerald, editor*: Rechtsfragen bei Open Source Software; Köln: Verlag Dr. Otto Schmidt KG, 2004, Print, ISBN 3–504–56080–0
*This book is based upon a legal opinion. Most of the chapters are written by Spindler himself and deal with the Open Source License types, the German Copyright ('Urheberrecht'), the German contract right ('Vertragsrecht') and the German liability law ('Haftungsrecht'). At the end it contains an excursus dealing with BSD and Mozilla licenses. In all cases the chapters deeply discuss different aspects of the topic.*

*Spindler, Gerald* a. *Andreas Wiebe*: Open Source-Vertrieb - Rechteeinräumung und Nutzungsberechtigung; in: Computerrecht, (2003), pp. 873–879
*[aus oberhem] [bibliographic data have be verified]*

*Splittgerber, Andrea; Schröder, Georg F., editor*: Lizenzen und Open Source rechtlicht einwandfrei nutzen. Eine klare Darstellung der Lizenzierung, Nutzungsrechtseinräumung und deren Auswirkung auf Vertragsgestaltung; Kissing: Weka Media, 2005, Print, ISBN 3–8245–1286–3
*The book describes the classical licenses detailedly. The Open Source part concerns only the GPL, on only a very few pages.*

*St. Laurent, Andrew N.*: Understanding open source and free software licensing: guide to navigating licensing issues in existing & new software; Beijing and Köln: O'Reilly, 2004
*next action, aus buchtala [bibliographic data have be verified]*

*Staff, CACM*: True seeds of open source software; in: Commun. ACM, 52 January (2009), pp. 6–6 ⟨URL: http://doi.acm.org/10.1145/1435417.1435420⟩, ISSN 0001–0782
*[bibliographic data have be verified]*

*Stahl, Matthew T.*: Open-source software: not quite endsville; in: Drug Discovery Today, 10 (2005), No. 3, pp. 219–222 ⟨URL: http://www.sciencedirect.com/science/article/pii/S1359644604033641⟩ – reference download: 2012-02-09, BibWeb/PDF
*[to be evaluated]*

*Stallman, Richard*: Viewpoint: Why We Must Fight UCITA; in: Communications of the ACM, 43 June (2000), No. 6, pp. 27–28 ⟨URL: http://doi.acm.org/10.1145/336460.336470⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Stallman, Richard*: Can Freedom Withstand E-Books? in: Communications of the ACM, 44 March (2001), No. 3, p. 111 ⟨URL: http://doi.acm.org/10.1145/365181.365227⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Stallman, Richard*: Viewpoint: Why "Open Source" Misses the Point of Free Software; in: Commununications of the ACM, 52 June (2009), No. 6, pp. 31–33 ⟨URL: http://doi.acm.org/10.1145/1516046.1516058⟩ – reference download: 2011-12-29, BibRef/PDF
*A newer refreshment of the elder position that open source software as label does not emphasize the idea of freedom: it refers to a development methodolgy while free software should be seen as a social movement.*

*Stallman, Richard M.*: The Danger of Software Patents; 2001, FreeWeb/HTML ⟨URL: http://www.gnu.org/philosophy/stallman-mec-india.html⟩ – reference download: 2013-02-18
*An early and elaborated description of possible effects and software patents and how to manage the corresponding challenges. [Lecture as given more than one time]*

*Stallman, Richard M.*: Can You Trust Your Computer? [originally written in 2002]; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 115–117, Print
*The article argues that 'DRM' - realized by a 'trusted computing' campaign - is technically established by a key based digital encryption of your data. This encryption is automatically incorporated into your computer (software). But because the keys are kept secret from the owner of the machine one can say that the programs using the hidden keys are controlling the owner, not the owner the computer. Therefore RMS uses the terms 'Digitial Restriction Management' and 'Treacherous Computing'.*

*Stallman, Richard M.*: The Danger of Software Patents; transcript of a speech given at University of Cambridge, London on the 25th of March 2002; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 95–111, Print
*Stallman emphasizes that a patent cover ideas and the use of ideas while copyrights only cover the details of expression of a work. Then he discussed approaches to react on a patent which tries to influence the development.*

*Stallman, Richard M.*: Free Software Definition; originally written in 1996; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 41–43, Print
*This article has defined the four constitutive features of free software: the freedom to run a program without any restrictions, the freedom to study how it works, the freedom to redistribute copies of the work and the freedom to improve it and to release the improvements. Additionally the article helps to interpret these features: The accessability to the sourcecode for example is a necessary condition, not a value itself. Or there might be free software which is not copyleft software.*

*Stallman, Richard M.; Gay, Joshua, editor*: Free Software, Free Society: Selected Essays of

Richard M. Stallman; [with an] Introduction by Lawrence Lessig; Boston, MA USA: GNU Press, 2002, Print, ISBN 1–882114–98–1

*A collection of those important articles by which RMS has established the philosophy of GNU.*

*Stallman, Richard M.*: Free Software: Freedom and Cooperation; transcript of a speech given at New York University on 29 May 2001; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 155–186, Print

*This transcript summarizes the main ideas of Free Software and their genesis: it tells the history of RMS.*

*Stallman, Richard M.*: Free Software Needs Free Documentation; originally written in 2000; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 67–68, Print

*The article argues that Free Software needs a good free documentation for being practically usable: the lack of good free manuals undermine the freedom to be able to use free software freely.*

*Stallman, Richard M.*: The GNU Manifesto; originally written in 1984; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 31–39, Print

*One of the earliest documents delineating the idea of the GNU project and GNU software. The larger part of the article discusses the question that free software offers advantages for all users, even in the context of a commercial use.*

*Stallman, Richard M.*: The GNU Project; originally published in 'Open Sources: Voices from the Open Source Revolution, O'Reilly, 1999'; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 15–30, Print

*This article summarizes aspects of the history, the targets, and the philosophy of GNU.*

*Stallman, Richard M.*: The Right to Read; originally written in 1997; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 73–77, Print

*From an anticipated viewpoint of the year 2096 Stallman analyzes a dilemma of a student who wants to help his girl friend. But he isn't allowed to help her because in this case it would be necessary to allow her to read his e-books stored on his computer. And that's forbidden and obstructed by a cooperation of the publisher and the computer companies.*

*Stallman, Richard M.*: Selling Free Software; originally written in 1996; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 63–65, Print

*This article opposes the myth that you should not charge money for distributing copies of free software. It underlines that - with one exception - the GPL has no requirements about how much you can charge for distributing a copy of free software: Following the GPL you can charge nothing, a penny, a dollar, or a billion dollars.*

*Stallman, Richard M.*: What is Copyleft? originally written in 1996; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 89–90, Print

*This article describes Copyleft as a method for making a programm free software and requiring all modified and extended versions of the program to be free software as well: firstly one states that the code is copyrighted, than - as copyright holder - one allows to use, modify, and redistribute the sourcecode under the condition, that this permission is not deleted.*

*Stallman, Richard M.*: What's in a Name? originally written in 2000; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 51–53, Print

*The articles emphasizes to talk about 'GNU/Linux' instead of 'Linux'. The simple name 'Linux' facilitates to combine free and proprietary software so that in the end the GNU campaign for freedom might have been failed. In the opposite the name 'GNU/Linux' preserves and transports the idea of the freedom to use free software in a specific manner, even to those who don't automatically agree with that philosophy.*

*Stallman, Richard M.*: Why 'Free Software' is Better than 'Open Software'; originally written in 1998; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 55–60, Print

*This article argues that the concept 'Open Source' was designed not to raise that users deserve freedom. And it underlines that the GNU project sticks to the term 'free software' for spreading the idea of freedom without any fear.*

*Stallman, Richard M.*: Why Software Should Not Have Owners; originally written in 1994; In *Stallman*: Free Software, Free Society: Selected Essays, 2002, pp. 45–49, Print

*This article repeats the statement that the society needs programs that people can read, fix, adapt, and improve. Instead of this software owners deliver black boxes that can't be studied. Additionally the article exemplifies some important methods to earn money with free software.*

*Stallman, Richard M.*: Let's Limit the Effect of Software Patents, Since We Can't Eliminate Them; in: Wired, n.st. January (2012), p. wp ⟨URL: `http://www.wired.com/opinion/2012/11/richard-stallman-software-patents/`⟩ – reference download: 2013-02-18, FreeWeb/HTML, ISSN n.st.

*Tries to solve the challenge of software patents by '[...] (legislating) that developing, distributing, or running a program on generally used computing hardware does not constitute patent infringement'.*

*Stallman, Richard M.*: Fighting Software Patents - Singly and Together; n.st. [2004], FreeWeb/HTML ⟨URL: `http://www.gnu.org/philosophy/fighting-software-patents.html`⟩ – reference download: 2013-02-18

*Deals with the fight against the planned new European patent law*

*Steinbring, Marc* a. *Thorsten Hampel*: Connecting Babbling Bazaars - Der Open-Source-Gedanke im Wandel zum offenen Service; In *Asche et al.*: Open Source. Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 73–97, Print

*The article addresses the 'symbiosis' of open structures and the commercial use. 'Open Source' seems to be taken as a spin doctor for the free WEB-2.0.*

*Stewart, Katherine J.*, *Anthony P. Ammeter*, a. *Likoebe M. Maruping*: Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects; in: Informations Systems Research, 17 (2006), No. 2, pp. 126–144, BibWeb/PDF

*[tbd: evaluate copy]*

*Stewart, Katherine J.*, *David P. Darcy*, a. *Sherae L. Daniel*: Observations on Patterns of Development in Open Source Software Projects; In Proceedings of the Fifth Workshop on Open Source Software Engineering, 2005, pp. 1–5 ⟨URL: `http://doi.acm.org/10.1145/1082983.1083272`⟩ – reference download: 2011-12-29, BibWeb/PDF

*[to be evaluated]*

*Stol, Klaas-Jan* a. *Muhammad Ali Babar*: Challenges in Using Open Source Software in Product Development: A Review of the Literature; [General Chairs: Justin Erenkrantz and Hyrum K. Wright]; In Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2010, pp. 17–22 ⟨URL: `http://doi.acm.org/10.1145/1833272.1833276`⟩ – reference download: 2011-12-29, BibWeb/PDF

*[to be evaluated]*

*Stol, Klaas-Jan et al.*: The Use of Empirical Methods in Open Source Software Research: Facts, Trends and Future Directions; In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009, pp. 19–24 ⟨URL: `http://dx.doi.org/10.1109/FLOSS.2009.5071355`⟩ – reference download: 2012-02-01, BibWeb/PDF

*[to be evaluated]*

*Subramaniam, Chandrasekar*, *Ravi Sen*, a. *Matthew L. Nelson*: Determinants of open source software project success: A longitudinal study; in: Decision Support System, 46 (2009), pp. 576–585, BibWeb/PDF
*[tbd: evaluate copy]*

*Subramanyam, Ramanath* a. *Mu Xia*: Free/Libre Open Source Software development in developing and developed countries: A conceptual framework with an exploratory study; in: Decision Support Systems, 46 (2008), No. 1, pp. 173–186 ⟨URL: `http://www.sciencedirect.com/science/article/pii/S016792360800119X`⟩ – reference download: 2012-02-03, BibWeb/PDF
*[to be evaluated]*

*Suchomski, Bernd*: Proprietäres Patentrecht beim Einsatz von Open Source Software. Eine rechtliche Analyse aus unternehmerischer Sicht; Bonn: Tgramedia, 2011 (= Medien Internet und Recht, [Vol./No.] 3), Print, ISBN 978–3–941192–03–4
*This book analyzes the meaning of 'gaining' and 'losing' a patent based on OSS: Firstly (in Germany) patents can be registers on the base of OSS. But mostly it's difficult to prevent their registration with a reference to an already existing OS software. Secondly in general a company distributes the right to use its' patents with distributing the OSS: In case of Non-Copyleft software the right is granted implicitly by granting the right to use the software. In case of copyleft software the patent must also be given free because of the obligation to publish the changings. And in case of OSS patent clauses all this is done explicitly.*

*Sujecki, Bartosz*: Open Source Software im deutschen Vetrags- und Urheberrecht; in: Medien und Recht, (2005), pp. 40–48
*[bibliographic data have be verified]*

*Sweet, David*: Andamooka: Open Support for Open Content; in: Linux Journal, 82 February (2001), pp. Article No. 13 ⟨URL: `http://dl.acm.org/citation.cfm?id=364716.364729`⟩ – reference download: 2011-12-28, BibWeb/HTML
*[to be evaluated]*

*Syme, Serena* a. *L. Jean Camp*: The Governance of Code: Open Land vs. UCITA Land; in: SIGCAS, 32 (2002), No. 3, p. 2 ⟨URL: `http://doi.acm.org/10.1145/644618.644623`⟩ – reference download: 2011-12-29, BibWeb/HTML
*[to be evaluated]*

*Taubert, Niels C.*: Produktive Anarchie? Netzwerke freier Softwareentwicklung; Bielefeld: transcript, 2006 (= Science Studies), Print, ISBN 3–89942–418–2
*This sociological book analyzes, in which sense the idea of 'Open Source' or 'Free Software' determines also the style of a cooperating elaboration. Roughly spoken this kind of collaboration must rather be characterized by argueing and making compromises than being determined by external decisions.*

*Terry, Michael*, *Matthew Kay*, a. *Ben Lafreniere*: Perceptions and Practices of Usability in the Free/Open Source Software (FoSS) Community; in: Proceedings of the 28th International Conference on Human Factors in Computing Systems; New York, NY, USA: ACM, 2010 (= CHI '10) ⟨URL: `http://doi.acm.org/10.1145/1753326.1753476`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–929–9, pp. 999–1008
*[to be evaluated]*

*Terry, Michael et al.*: ingimp: Introducing Instrumentation to an End-User Open Source Application; in: Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems; New York, NY, USA: ACM, 2008 (= CHI '08) ⟨URL: `http://doi.acm.org/10.1145/1357054.1357152`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–60558–011–1, pp. 607–616
*[to be evaluated]*

*Teupen, Christian*: 'Copyleft' im deutschen Urheberrecht; Implikationen von Open Source Software im Urhebergesetz; Berlin: Duncker & Humblot, 2007 (= Schriften zum Bürgerlichen Recht, [Vol./No.] 367), Print, ISBN 978–3–428–12325–4
*[tbd: evaluate copy]*

*Themelidis, Markos*: Open Source : die Freiheitsvision der Hacker; Frankfurt a.M.: ???, 2004
*Diplomarbeit, Universität Frankfurt, BGE, Abteilung Gesellschaftswissenschaften, gg 600 ; lx 130 EXAM 1152, Kurzausleihe verfuegbar Magazinbestand, bitte vor Ort bestellen [bibliographic data have be verified]*

*Theunissen, W. H. Morkel*, *Andrew Boake*, a. *Derrick G. Kourie*: In Search of the Sweet Spot: Agile Open Collaborative Corporate Software Development; in: Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries; Republic of South Africa: South African Institute for Computer Scientists and Information Technologists, 2005 (= SAICSIT '05) ⟨URL: `http://dl.acm.org/citation.cfm?id=1145675.1145705`⟩ – reference download: 2011-12-28, BibWeb/PDF, ISBN 1–59593–258–5, pp. 268–277
*[to be evaluated]*

*Thorvalds, Linus*: Just for fun : wie ein Freak die Computerwelt revolutionierte; die Biographie des Linux-Erfinders; München: ???, 2004
*Präsenzbestand: Universität Frankfurt, Bibliothekszentrum Geisteswissenschaften 14/X. a. 30 verfuegbar IG Farben-Haus, Q1, 5. OG, Raum 5.111 [bibliographic data have be verified]*

*Torkar, Richard*, *Pau Minoves*, a. *Janina Garrigós*: Adopting Free/Libre/Open Source Software. Practices, Techniques and Methods for Industrial Use; in: JAIS, 12 (2011), No. 1, pp. 88–122, BibWeb/PDF
*[to be evaluated]*

*Torvald, Linus*: Just for Fun: wie ... [Biographie]; ???, 2004
*[Präsenz FaM IG-Farbenhaus Q1 5 QG Raum 5.111 14/X a.30] [bibliographic data have be verified]*

*Tsai, John*: For Better or Worse: Introducing the GNU General Public License Version 3; in: Berkeley Technology Law Review, 23 (2008), pp. 547–581, Copy
*FernleihCopy*

*Tsunoda, Masateru et al.*: Analyzing OSS Developers' Working Time Using Mailing Lists Archives; in: Proceedings of the 2006 International Workshop on Mining Software Repositories; New York, NY, USA: ACM, 2006 (= MSR '06) ⟨URL: `http://doi.acm.org/10.1145/1137983.1138031`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 1–59593–397–2, pp. 181–182
*[to be evaluated]*

*Turner, David*: Anatomy of GPL Violations; in: Free Software Foundation Bulletin, (2002), No. 1, pp. 2–3
*[aus Widmer] [bibliographic data have be verified]*

*Tuunanen, Timo*, *Jussi Koskinen*, a. *Tommi Kärkkäinen*: Automated software license analysis; in: Automated Software Engineering, 16 (2009), pp. 455–490, BibWeb/PDF
*[tbd: evaluate copy]*

*Twidale, Michael*: Silver Bullet or Fool's Gold: Supporting Usability in Open Source Software Development; in: Proceedings of the 27th International Conference on Software Engineering; New York, NY, USA: ACM, 2005 (= ICSE '05) ⟨URL: `http://doi.acm.org/10.1145/1062455.1062468`⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 1–58113–963–2, p. 35
*[to be evaluated]*

*Uhr, Wolfrgang*, *Werner Esswein*, a. *Eric Schoop, editors*: Wirtschaftsinformatik 2003 / Band II. Medien - Märkte - Mobilität; Heidelberg: Physica-Verlag, 2003

*next action, aus buchtala [bibliographic data have be verified]*

*Välimäki, Mikko*: Copyleft Licensing and EC Competition Law; in: E.C.L.R, 27 (2006), No. 3, pp. 130–136, Copy
*Fernleih*

*Välimäki, Mikko*, *Ville Oksanen*, a. *Juha Laine*: An Empirical Look at the Problems of Open Source Adoption in Finnish Municipalities; in: Proceedings of the 7th International Conference on Electronic Commerce; New York, NY, USA: ACM, 2005 (= ICEC '05) ⟨URL: `http://doi.acm.org/10.1145/1089551.1089643`⟩ – reference download: 2011-12-29, ISBN 1–59593–112–0, pp. 514–520
*[to be evaluated]*

*Valkov, Svilen*: Innovative Concept of Open Source Enterprise Resource Planning (ERP) System; in: Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing; New York, NY, USA: ACM, 2008 (= CompSysTech '08) ⟨URL: `http://doi.acm.org/10.1145/1500879.1500893`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–954–9641–52–3, pp. 11.6.1–11.6.7
*[to be evaluated]*

*Vamplew, Peter* a. *Julian Dermoudy*: An Anti-Plagiarism Editor for Software Development Courses; in: *Alison Young* a. *Denis Tolhurst, editors:* Australasian Computing Education Conference; Necastle (Australia), 2005 (= Australia. Conferences in Researhc an Practice in Information Technology, [Vol./No.] 42), BibWeb/PDF, pp. 83–90
*[to be evaluated]*

*Van den Brande, Ywain*, *Shane Coughlan*, a. *Till Jaeger, editors*: The International Free and Open Source Software Law Book; Munich (Germany): Open Source Press, 2011, Print, ISBN 978–3–941841–49–9
*Beside a very short historical introduction this book contains legal descriptions of the Open Source context and interpretation in different European countries.*

*van Wendel de Joode, R.*, *J. A. de Bruijn*, a. *M. J. G. van Eeten*: Protecting the Virtual Commons. Self-Organizing Open Source and Free Software Communities and Innovative Intellectual Property Regimes; The Hague: T.M.C. Asser Press, 2003 (= Information Technology & Law Series, [Vol./No.] 3), Print, ISBN 90–6704–159–9
*Beside other questions this report wants to answer the question why the communities have created so many different Free / Open Source Software licenses. The answer is perhaps a little thin: the most part of the licenses are derived from BSD or GPL 'to protect the fundamental principles of the communities'.*

*Ven, Kris* a. *Jan Verelst*: The Importance of External Support in the Adoption of Open Source Server Software; conference contribution; In *Boldyreff et al.*: Open Source Ecosystems, 2009, pp. 116–128, BibWeb/PDF
*[tbd: evaluate copy]*

*Vetter, Greg R.*: 'Infectious' Open Source Software: Spreading Incentives or Promoting Resistance? in: Rutgers Law Journal, 36:53 (2005), pp. 53–162
*next action, aus buchtala [bibliographic data have be verified]*

*Viesel, Edward*: Freiheit statt Freibier. Geschichte und Praxis der freien digitalen Welt - mit einer Einführung in Linux; Münster: Unrast-Verlag, 2006, Print, ISBN 3–897771–450–7
*One half of the book introduces into GNU/Linux, nearly a quarty into DRM, formats, and linked collaborative projects. And the first quarter summarizes the history of Open Source and some ideas of OS licenses.*

*von Hippel, Eric* a. *Georg von Krogh*: Open source software and the private-collective innovation model: Issues for organization science; in: Organization Science, 14 (2002), No. 2, pp. 209–223
*next action, aus buchtala [bibliographic data have be verified]*

*von Hippel, Eric* a. *Georg von Krogh*: The Promise of Research on Open Source Software; in: Management Science, 52 (2006), No. 7, pp. 975–983
*next action, aus buchtala [bibliographic data have be verified]*

*von Krogh, G.* a. *E. von Hippel*: The Promise of Research on Open Source Software; in: Management Science, 2006 (52), pp. 975–983
*[bibliographic data have be verified]*

*von Krogh, Georg* a. *Sebastian Spaeth*: The open source software phenomenon: Characteristics that promote research; in: Journal of Strategic Information Systems, 16 (2007), p. 236–253, BibWeb/PDF
*[to be evaluated]*

*Välimäki, Mikko*: The Rise of Open Source Licensing; A Challenge to the Use of Intellectual Property in the Software Industry;, PhD thesis ⟨URL: `http://pub.turre.com`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Välimäki, Mikko*: Dual Licensing in Open Source Software Industry; 2003 ⟨URL: `http://opensource.mit.edu/papers/valimaki.pdf`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Wang, Yi*, *Defeng Guo*, a. *Huihui Shi*: Measuring the Evolution of Open Source Software Systems with their Communities; in: SIGSOFT Software Engineering Notes, 32 November (2007), No. 6, pp. 1–7 ⟨URL: `http://doi.acm.org/10.1145/1317471.1317479`⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Watson, Richard T. et al.*: The Business of Open Source; in: Communications of the ACM, 51 (2008), No. 4, pp. 41–46 ⟨URL: `http://doi.acm.org/10.1145/1330311.1330321`⟩
*[to be evaluated]*

*Weber, S.*: The Success of Open Source; Cambridge MA: Harvard University Press, 2004
*[bibliographic data have be verified]*

*Weiss, Aaron*: The Politics of Free (Software); in: netWorker, 5 September (2001), pp. 26–31 ⟨URL: `http://doi.acm.org/10.1145/383719.383727`⟩
*[to be evaluated]*

*Weiss, Michael*: Economics of Collectives; in: Proceedings of the 15th International Software Product Line Conference; Volume 2, New York, NY, USA: ACM, 2011 ⟨URL: `http://doi.acm.org/10.1145/2019136.2019181`⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0789–5, pp. 39:1–39:8
*[to be evaluated]*

*West, Joel*: How open is open enough? Melding proprietary and open source platform strategies; in: Research Policy, 32 (2003), pp. 1259–1285
*next action, aus buchtala [bibliographic data have be verified]*

*Wheeler, David A.*: Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers! 2002 ⟨URL: `http://www.dwheeler.com/oss_fs_why.html`⟩
*next action, aus buchtala [bibliographic data have be verified]*

*Wichmann, Thorsten*: Linux- und Open-Source-Strategien; Berlin, Heidelberg and New York: Springer, 2005, BibWeb/PDF, ISBN 3–540–22810–1
*[tbd: evaluate copy]*

*Widmer, Mike J.*: Open Source Software - Urheberrechtliche Aspekte freier Software; Dissertation; Bern: Stämpfli Verlag, 2003, Print
*The book explains concepts of Open Source movement and its history competently. It also describes 'Non Copyleft Licenses', 'Copyleft Licenses' and other forms like proprietary or commercial licenses etc. Then it analyzes systematical connections between Open Source Software and the (Europaen/Swiss) 'Urheberrecht' (Copyright). And finally it analyzes - clause by clause - the links from the GPL into the 'Urheberrecht' and vice versa.*

*Wiebe, A.*: Softwarepatente und Open Source; in: CR [Computer und Recht], 20 (2004), No. 12, pp. 881–888
*[aus Asche u. Bauhus] [bibliographic data have be verified]*

*Wikipedia (de)*: Microsoft Public License; n.l, 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://de.wikipedia.org/wiki/Microsoft_Public_License`⟩ – reference download: 2013-02-26
*German explanation of the MS-PL which erroneously says that the Microsft Public license evokes a weak copyleft effect.*

*Wikipedia (de)*: Microsoft Reciprocal License; n.l, 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://de.wikipedia.org/wiki/Ms-RL`⟩ – reference download: 2013-02-26
*German explanation of the MS-RL which correctly describe its weak copyleft effect.*

*Wikipedia (en)*: Free and open source software; n.l., 2011, FreeWeb/HTML (German Version unter http://de.wikipedia.org/wiki/FLOSS) ⟨URL: `http://en.wikipedia.org/wiki/Free_and_open_source_software`⟩ – reference download: 2011-09-08
*Illuminates the attempt to (re-)amalgamate the split Free Software movement and Open Source movement at least on the level of meta-concepts by simply linking the majuskels of Free, Libre, Open, Source, Software as hybrid concepts. 'Libre' is introduced to resolve the ambiguity of 'free' in the sense of 'freedom'.*

*Wikipedia (en)*: MIT License; n.l, 2011, FreeWeb/HTML ⟨URL: `http://en.wikipedia.org/wiki/MIT_License`⟩ – reference download: 2011-09-20
*The article explains clearly that there doesn't exist one single MIT-License: one must distinguish between the simpler MIT-expat-License and the more complex MIT-X11-License. But basically the two license denote the same license class. They protect the developer, not the code.*

*Wikipedia (en)*: Copyleft; n.l., 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://en.wikipedia.org/wiki/Copyleft`⟩ – reference download: 2013-02-02
*Describes and deliniates strong and weak copyleft*

*Wikipedia (en)*: Permissive free software licence; n.l., 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://en.wikipedia.org/wiki/Permissive_free_software_licence`⟩ – reference download: 2013-02-02
*May be read as an example for the use of the concept 'permissive free software licenses'.*

*Wikipedia (en)*: Shared source; n.l, 2013 [n.y.], FreeWeb/HTML ⟨URL: `http://en.wikipedia.org/wiki/Shared_source`⟩ – reference download: 2013-02-26
*Brief summary of the different license models of Microsoft which also explains the difference between the permissive license MS-PL and the weak copyleft license MS-RL.*

*Williams, Sam*: Free as in Freedom. Richard Stallman's Crusade for Free Software; Beijing [... etc.]: O'Reilly, 2002, Print, ISBN 0–596–00287–4
*This is the first(?) biography of RMS. It contains so many interviews and background information that it might be read as primary source, not only for the history of RMS, but also for the history of the Free Software movement and its internal counterpart, the Open Source movement.*

*Witzel, Michaela*: AGB-Recht und Open Source Lizenzmodelle; in: ITRB (IT-Rechtsberater), (2003), pp. 175ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Wolf, M.*, *K. Miller*, a. *F. Grodzinsky*: On the meaning of free software; in: Ethics and Information Technology, 11 (2009), pp. 279–286 ⟨URL: `http://dx.doi.org/10.1007/s10676-009-9207-9`⟩ – reference download: 2012-02-03, BibWeb/PDF
*[to be evaluated]*

*Wolf, Marty J. et al.*: Open Source Software: Intellectual Challenges to the Status Quo; in: Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education;

New York, NY, USA: ACM, 2002 (= SIGCSE '02) ⟨URL: http://doi.acm.org/10.1145/563340.563464⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 1–58113–473–8, pp. 317–318
*[to be evaluated]*

*Wolf, Marty J.*, *Keith W. Miller*, a. *Frances S. Grodzinsky*: Free, Source-Code-Available, or Proprietary: An Ethically Charged, Context-Sensitive Choice; in: SIGCAS, 39 June (2009), No. 1, pp. 15–26 ⟨URL: http://doi.acm.org/10.1145/1565795.1565797⟩
*[to be evaluated]*

*Wolfe, Alexander*: Toolkit: GNU Tools: Still Relevant? in: Queue, 1 December / January (2003/2004), pp. 14–17 ⟨URL: http://doi.acm.org/10.1145/966789.966795⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Wuermeling, Ulrich* a. *Thies Deike*: Open Source Software: Eine juristische Risikoanalyse; in: CR [Computer und Recht], (2003), pp. 87ff
*aus Koeglin2007a [bibliographic data have be verified]*

*Wynants, Marleen* a. *Jan Cornelius, editors*: How Open is the Future? Economic, Social & Cultural Scenarios inspired by Free & Open-Source Software; Brüssel: VUB Brussels University Press, 2005
*next action, aus buchtala [bibliographic data have be verified]*

*Xing, Guangming*: Teaching Software Engineering Using Open Source Software; in: Proceedings of the 48th Annual Southeast Regional Conference; New York, NY, USA: ACM, 2010 (= ACM SE '10) ⟨URL: http://doi.acm.org/10.1145/1900008.1900085⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0064–3, pp. 57:1–57:3
*[to be evaluated]*

*Xu, Bo*, *Donald R. Jones*, a. *Bingjia Shao*: Volunteers' involvement in online community based software development; in: Information & Management, 46 (2009), pp. 151–158, BibWeb/PDF
*[to be evaluated]*

*Yamakami, Toshihiko*: Foundation-based Mobile Platform Software Engineering: Implications to Convergence to Open Source Software; in: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human; New York, NY, USA: ACM, 2009 (= ICIS '09) ⟨URL: http://doi.acm.org/10.1145/1655925.1655962⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–60558–710–3, pp. 206–211
*[to be evaluated]*

*Yamakami, Toshihiko*: OSS as a digital ecosystem: A Reference Model for Digital Ecosystem of OSS; in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems; New York, NY, USA: ACM, 2010 (= MEDES '10) ⟨URL: http://doi.acm.org/10.1145/1936254.1936291⟩ – reference download: 2011-12-29, BibWeb/PDF, ISBN 978–1–4503–0047–6, pp. 207–208
*[to be evaluated]*

*Yatani, Koji et al.*: Understanding How and Why Open Source Contributors Use Diagrams in the Development of Ubuntu; in: Proceedings of the 27th International Conference on Human Factors in Computing Systems; New York, NY, USA: ACM, 2009 (= CHI '09) ⟨URL: http://doi.acm.org/10.1145/1518701.1518853⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–60558–246–7, pp. 995–1004
*[to be evaluated]*

*Ye, Yunwen* a. *Kouichi Kishida*: Toward An Understanding of the Motivation of Open Source Software Developers; in: Proceedings of the 25th International Conference on Software Engineering; Washington, DC, USA: IEEE Computer Society, 2003 (= ICSE '03) ⟨URL:

http://dl.acm.org/citation.cfm?id=776816.776867⟩, ISBN 0–7695–1877–X, pp. 419–429
*[to be evaluated]*

*Yildirim, Nihan* a. *Hacer Ansal*: Foresighting FLOSS (free/libre/open source software) from a developing country perspective: The case of Turkey; in: Technovation, 31 (2011), No. 12, pp. 666 – 678 ⟨URL: http://www.sciencedirect.com/science/article/pii/S0166497211001052⟩, ISSN 0166–4972
*[bibliographic data have be verified]*

*Yue, Kwok-Bun et al.*: The Use of Free and Open Source Software in Real-World Capstone Projects; in: JCSC, 26 April (2011), No. 4, pp. 85–92 ⟨URL: http://dl.acm.org/citation.cfm?id=1953573.1953587⟩ – reference download: 2011-12-29, BibWeb/PDF
*[to be evaluated]*

*Yue, Kwok-Bun et al.*: Open Courseware and Computer Science Education; in: JCSC, 20 October (2004), No. 1, pp. 178–186 ⟨URL: http://dl.acm.org/citation.cfm?id=1040231.1040255⟩
*[to be evaluated]*

*Zacchiroli, Stefano*: Debian: 18 years of free software, do-ocracy, and democracy; in: Proceedings of the 2011 Workshop on Open Source and Design of Communication; New York, NY, USA: ACM, 2011 (= OSDOC '11) ⟨URL: http://doi.acm.org/10.1145/2016716.2016740⟩, ISBN 978–1–4503–0873–1, pp. 87–87
*[bibliographic data have be verified]*

*Zhang, Wen*, *Ye Yang*, a. *Qing Wang*: Network Analysis of OSS Evolution: An Empirical Study on ArgoUML Project; in: Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution; New York, NY, USA: ACM, 2011 (= IWPSE-EVOL '11) ⟨URL: http://doi.acm.org/10.1145/2024445.2024459⟩ – reference download: 2012-02-01, BibWeb/PDF, ISBN 978–1–4503–0848–9, pp. 71–80
*[to be evaluated]*

*Zhou, Ying* a. *Joseph Davis*: Open source software reliability model: an empirical approach; in: Proceedings of the fifth workshop on Open source software engineering; New York, NY, USA: ACM, 2005 (= 5-WOSSE) ⟨URL: http://doi.acm.org/10.1145/1082983.1083273⟩, ISBN 1–59593–127–9, pp. 1–6
*[bibliographic data have be verified]*

*Zittrain, Jonathan*: Normative Principles for Evaluating Free and Proprietary Software; in: University of Chicago Law Review, 71 (2004), No. 1, pp. 265–287, BibWeb/PDF
*[to be evaluated]*

*Zucker, William A.*: Intellectual Property and Open Source: Copyright, Copyleft, and Other Issues for the User Community; in: Cutter IT Journal, 16 (2003), No. 5, pp. 27–34, Copy *Fernleihkopie*