

Release 0.1.98

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

Open Source License Compendium

How to to Use Open Source Software in a Regular Manner

K. Reincke, J. Dobson, G. Sharpe, P. Schichl, M. Kern*

3rd March 2012

* 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 quotations please use the scientific standard to cite.

[derived from myCsrF (= 'mind your Scholar Research Framework') ©K. Reincke CC BY 3.0 <http://mycsrf.fodina.de/>]

The Open Source Community is a swarm: it is stronger than a set of some accidentally selected experts. We are proud to get its' help. Gladly we thank (in alphabetical order):

Michael Kern,
Thomas Quiehl,
Helene Tamer,
and all the
other...

Contents

1	Introduction	7
2	Prolegomena	8
2.1	Why	8
2.2	What	16
3	Open Source: Idea and Concepts	20
3.1	Open Source, OSI, and OSD	20
3.2	Open Source and its history: some hints	20
3.3	<i>Free Software</i> versus <i>Open Source</i> : some hints	21
4	The Same Idea, Different Models of License	22
4.1	Minimal Prescribing Licensemodels: Protecting the developer	22
4.2	Reflexive Prescribing Licensemodels: Protecting the code	22
4.3	Overlapping Prescribing Licensemodels: Protecting the on-top-developments	22
4.3.1	Excursion: GPL is not the Evil	23
4.3.2	Excursion: The Peculiarity of GPL-3	23
4.4	Other Models	23
4.5	Summary: Our Taxonomy for the analyzed Licenses	23
5	Open Source: Important Minor Points	24
5.1	Excursion: The Problem of Derivated Works	24
5.2	Excursion: The Problem of Combining Different Licensemodels	24
5.3	Excursion: Open Source Software and Money	24
5.4	Excursion: The Problem of Implicitly Freeing Patents	25
6	Open Source: Use Cases As Principle of Classification	26
7	Open Source Licenses: Find Your Specific To-do Lists	27
8	Open Source License Fulfillment: Classified To-do Lists for . . .	28
8.1	Apache Licensed Software in the usage context of	28
8.1.1	Without Redistribution and	28
8.1.1.1	As-It-Is-Usage	28
8.1.1.2	With-Code-Modification	28
8.1.1.3	As-Isolated-Initial-Development	28
8.1.1.4	As-On-Top-Development	28
8.1.2	With Redistribution and	28
8.1.2.1	As-It-Is-Usage	28
8.1.2.2	With-Code-Modification	29
8.1.2.3	As-Isolated-Initial-Development	29
8.1.2.4	As-On-Top-Development	29
8.2	etc.	29

Contents

8.3	BSD Licensed Software in the usage context of ...	29
8.3.1	Without Redistribution and ...	29
8.3.1.1	As-It-Is-Usage	29
8.3.1.2	With-Code-Modification	29
8.3.1.3	As-Isolated-Initial-Development	29
8.3.1.4	As-On-Top-Development	29
8.3.2	With Redistribution and ...	30
8.3.2.1	As-It-Is-Usage	30
8.3.2.2	With-Code-Modification	30
8.3.2.3	As-Isolated-Initial-Development	30
8.3.2.4	As-On-Top-Development	30
8.4	GPL V2 Licensed Software in the usage context of ...	30
8.4.1	Without Redistribution and ...	30
8.4.1.1	As-It-Is-Usage	30
8.4.1.2	With-Code-Modification	30
8.4.1.3	As-Isolated-Initial-Development	30
8.4.1.4	As-On-Top-Development	30
8.4.2	With Redistribution and ...	31
8.4.2.1	As-It-Is-Usage	31
8.4.2.2	With-Code-Modification	31
8.4.2.3	As-Isolated-Initial-Development	31
8.4.2.4	As-On-Top-Development	31
8.5	etc.	31
8.6	MIT Licensed Software in the usage context of ...	31
8.6.1	Without Redistribution and ...	31
8.6.1.1	As-It-Is-Usage	31
8.6.1.2	With-Code-Modification	31
8.6.1.3	As-Isolated-Initial-Development	31
8.6.1.4	As-On-Top-Development	31
8.6.2	With Redistribution and ...	32
8.6.2.1	As-It-Is-Usage	32
8.6.2.2	With-Code-Modification	32
8.6.2.3	As-Isolated-Initial-Development	32
8.6.2.4	As-On-Top-Development	32
9	Open Source Licenses and Their Legal Environments	33
10	Conclusion	34
11	Appendices	35
11.1	Some Widespread Open Source Myths	35
	Periodicals, Shortcuts, and Overlapping Abbreviations	37
	Bibliography	39

Contents

Table 0.1: History of the Open Source License Compendium

2012-01-31	0.1.8	▷ renamed existing introduction as prolegomena ▷ inserted a shorter top-down written introduction ▷ inserted an OSLiC disclaimer
2012-01-21	0.1.7	▷ oscCopiedButNotRead.bib expanded by many verified bibliographic data ▷ list of periodicals and shortcuts added
2011-12-29	0.1.6	▷ many bibliographic data added
2011-10-17	0.1.5	▷ bibliographic data updated
2011-09-29	0.1.4	▷ Document history integrated ▷ Some Typos erased
2011-09-28	0.1.3	▷ Review of english teacher integrated
2011-09-19	0.1.2	▷ First comments of english teacher integrated
2011-09-15	0.1.1	▷ Improvements of John integrated
2011-09-12	0.1.0	▷ Introduction completed: purpose and methods

Disclaimer

This book shall be thoroughly developed - together with the Open Source Community. Finally it shall deliver reliable information, following the rule, that the swarm knows more than the single fish.

But nevertheless it can't offer more than the opinion(s) of its authors and contributors. It's only one voice of chorus discussing the topic of 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 PROGRAM (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 PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1 Introduction

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.

[TDB ...]

2 Prolegomena

This chapter describes why we need an OSLiC and how its content and form has been derivated from the need. It's written as bottom-up explanation and shall deliver deeper insights.

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 is 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 incorrect¹.

¹ 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.

2 Prolegomena

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 can not 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 who were known as Open Source enthusiasts - to establish a service and a process for answering this question.

So, it is no surprising 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:

Firstly, we already knew that there exist supporting software. These meta-programs take the code of any other application and try to list those Open Source Software being 'covered' by that application². But we had also already realised that this supporting software did not always match the way we thought the problem should be solved. Secondly 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 reliable than this. 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

² As general examples let us mention Palamida (<http://www.palamida.com/>) and BlackDuck (<http://www.blackducksoftware.com/>).

2 Prolegomena

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 relating to individual issues spread over many sources. We did not find answers for 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's one of the most important pieces of Open Source Software³ with a specific license⁴. 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*⁵. 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 together⁶. Hence many business models depends on the Apache License. Another aspect is that even the famous *Apache Cookbook*, which explains the installation, the configuration and the maintaining of an Apache Web Server detailedly⁷, does not mention anything about

³ To prove that the *Apache* is really a piece of Open Source Software one must execute a set of steps: Firstly 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.). Therefor 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 subsite* (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]

⁴ 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..

⁵ cf. *Moody, Glyn*: Die Software-Rebellen. Die Erfolgsstory von Linus Torvalds und Linux; aus dem Amerikanischen übers. v. Annemarie Pumpering; Amerikanische Ausgabe von aus dem Jahr 2000; Landsberg am Lech: verlag moderne industrie, 2001, ISBN 3-478-38730-2, pp. 287ff.

⁶ 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.

⁷ cf. *Coar, Ken a. Rich Bowen*: Apache Kochbuch; deutsche Übersetzung v. Jochen Wiedmann; Beijing [... u.a.O.]: O'Reilly, 2004, ISBN 3-89721-371-0, et passim.

2 Prolegomena

the license which allows for installation, configuration and maintenance. Neither the index lists the word 'license'⁸, nor the chapters 'Installation'⁹ or the chapter 'Miscellaneous'¹⁰ 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 it¹¹. 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's 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 difficult¹². So again: Can you find your answer in another book, which is *Amazon's* current top recommendation for the request 'apache server'¹³? Not really: Sascha Kersken's Apache 2.2 Handbook offers a license chapter, but only two pages long¹⁴. Moreover the rights and duties are condensed into just 5 bullet points which taken together do not explain when the software and the license has to be handed over to a customer and when you are allowed to hide your improvements¹⁵.

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*¹⁶, founded in 2000 as a private institute to track the phenomenon 'free software' from the viewpoint of (German) lawyers¹⁷. Besides

⁸ cf. *Coar a. Bowen*: Apache Kochbuch, 2004, pp. 245ff, esp. p. 250.

⁹ cf. *id.*, l.c., pp. 1ff.

¹⁰ cf. *id.*, l.c., pp. 219ff.

¹¹ 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“.

¹² 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?

¹³ Tested on <http://www.amazon.de/> at 2011-08-31.

¹⁴ cf. *Kersken, Sasche*: Apache 2.2. Das umfassende Handbuch; 3., aktualisierte u. erw. Aufl.; Bonn: Galileo Press, 2009, ISBN 978-8362-1325-7, pp. 111f.

¹⁵ cf. *id.*, l.c., p. 112.

¹⁶ originally: „Institut für Rechtsfragen der Freien und Open Source Software“. Main entry point for its' site is the URL <http://www.iffross.org/>.

¹⁷ cf. *iffross*: Ziele, Aufgaben, Geschichte; 2011 (URL: <http://www.iffross.org/node/16>) – reference download: 2011-09-05, wp.

2 Prolegomena

many other aspects this site offers a very well and thoroughly elaborated FAQ¹⁸ 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)¹⁹.

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 bibliographies²⁰: the book of Jaeger and Metzger which concerns - loosely translated - *the judicial framework requirement for Open Source Software*²¹. 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 duties²² of the software user²³. Many other important aspects of the topic *Open Source* are discussed, too²⁴.

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, either in the first edition²⁵ and in the second edition²⁶ or in the recently published third edition²⁷. So, how can a company or a developer or a project manager be 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 '*then do this*'? Why should he himself implicitly become an Open Source Licenses expert which has to extract the necessary steps out of the literature?

¹⁸ cf. *ifross*: FAQ - Häufig gestellte Fragen; 2011 (URL: <http://www.ifross.org/faq-haeufig-gestellte-fragen>) – reference download: 2011-09-05, wp.

¹⁹ cf. *ifross*: Lizenz-Center; 2011 (URL: <http://www.ifross.org/lizenz-center>) – reference download: 2011-09-05, wp.

²⁰ at least in that German judicial literature dealing with Open Source

²¹ 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.).

²² cf. id., l.c., pp. 30ff.

²³ For getting a good survey of the structure and the line of thought see the contents cf. id., l.c., pp. VIIIff.

²⁴ pars pro toto: have a look at the chapter concerning the liability: cf. id., l.c., pp. 137ff.

²⁵ cf. id., l.c., pp. VIff.

²⁶ 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. VIIIff.

²⁷ cf. *Jaeger, Till a. Axel Metzger*: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 3rd edition. München: Verlag C.H. Beck, 2011, pp. VIIIff. Naturally we use this latest edition for adopting or discussing systematical aspects.

2 Prolegomena

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'²⁸. It aims to be a 'pragmatic guidebook' and an 'assistance' for 'publishing software under the conditions of an Open Source License'²⁹. 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 software³⁰.

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 public³¹. 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 author³². Moreover, besides other aspects the articles discusses briefly and deeply 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 donation³³. And last but not least this article actually summarizes the idea of Copyleft and the differences between LGPL and GPL³⁴.

However some gaps remain. The article does not analyze in which cases a University or a company perhaps *must* publish its' developments based upon 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 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.

²⁸ approximately translated

²⁹ 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“).

³⁰ 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).

³¹ cf. *id.*, l.c., pp. 170 and 181.

³² cf. *id.*, l.c., p. 173.

³³ cf. *id.*, l.c., p. 177.

³⁴ cf. *id.*, l.c., p. 181.

2 Prolegomena

We did, however, feel supported by this article, in two ways. Firstly it was a well written summary of some main problems. Secondly it stated the necessity to have a compendium for being able to establish a legally 'water-tight' process of publishing Open Source software³⁵. 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 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-Licence-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 without question. We wanted to be revisable. Because our sources and our conclusions should be easily verifiable, we decided to use the academic citations and list bibliographic data extensively on the basis

³⁵ cf. *Bretschneider, Glaschick, a. Gräfe*: Ratgeber für die Veröffentlichung von Open-Source-Software durch eine Hochschule, 2008a, p. 186.

2 Prolegomena

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 system³⁶), we wanted to take them as our guideline - at least while they do not violate more general laws³⁷. 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 anything³⁸, we would interpret the open issues in the spirit of the Open Source idea. But where the text was clear and

³⁶ 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.*

³⁷ what they clearly do not do!

³⁸ 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 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

2 Prolegomena

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 license³⁹, 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.

2.2 What

Now we can briefly explain how you can use this compendium:

Open Source: Idea and Concepts :- Here you will find background information to help you interpret Open Source Licenses in the sense of the *Free Software*

GPL; KR.) in design almost achieves what it has to achieve“. Hence we are asked to take the license texts themselves seriously. cf. *Williams, Sam*: Free as in Freedom. Richard Stallman's Crusade for Free Software; Beijing [... a.a.O.]: O'Reilly, 2002, ISBN 0-596-00287-4, pp. 177f.

³⁹This 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 quotations please use the scientific standard to cite.

[derived from *myCsrF* (= 'mind your Scholar Research Framework') ©K. Reincke CC BY 3.0 <http://mycsr.fodina.de/>]

2 Prolegomena

*movement*⁴⁰, the *Open Source Software movement*⁴¹ or the GNU-Project⁴². If you are familiar with the evolution of the Open Source Initiative, with the character of the Open Source Definition as a set of necessary but insufficient criteria and if you know about the history and meaning of free software as older and broader concept then you can ignore this chapter.

The Same Idea, Different Licenses :- In this chapter we discuss different ways to cluster Open Source Licenses. Finally we present our own taxonomy based on the labels 'protecting the developer', '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 than it's 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 developer 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's 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 they are handled by some Open Source

⁴⁰ 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 (de)*: Free/Libre Open Source Software; n.l., 2011 (URL: <http://de.wikipedia.org/wiki/FLOSS>) – reference download: 2011-09-08, wp..

⁴¹ For another brief and informative introduction 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, pp. 231ff esp. p. 232f..

⁴² 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.

2 Prolegomena

Licenses. You should know what licenses implicitly do with your patents. But it's 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 that chapter which you probably will reuse multiply, 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 that 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.

A final remark: We have already characterized the tone of our footnotes. 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 -

2 Prolegomena

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? Firstly 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. Secondly we store all these e-wares⁴³ we read⁴⁴. 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 data⁴⁵:

- 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 url⁴⁶ and the date when we downloaded / saw the text.

⁴³Take this little word as (new) generalization of 'e-book', 'e-article', 'e-paper' and so on.

⁴⁴But 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.

⁴⁵And another hint: Nowadays sometimes even scientific libraries doesn't offer exact 'e-copies' of the original. In some cases one can get only 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.

⁴⁶Please 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 sometime split the urls and inserted blanks. So you have to erase all blanks if you want to verify our urls.

3 Open Source: Idea and Concepts

In this chapter we discuss the meaning of Open Source and the common features of all Open Source Software. We scan historical and conceptual aspects. It's the chapter of background knowledge. At the end you will know that software is only Open Source Software if you as customer gets the right to use, to modify, and to redistribute the code without any limitations, but with some obligations. And these obligations implicitly refer to different historical contexts which might influence the understanding of your licenses.

[TDB ...]

3.1 Open Source, OSI, and OSD

Here we describe the meaning of Open Source. At the end of this chapter you should know that Open Source Software is defined by a set of necessary criteria which together determine the common basic features of Open Source Licenses. Additionally you will have understood that the opposite of Open Source Software can not only be defined ex negativo. But you should also know that these features can differently be implemented. Therefore the OSD can not be read as a set of rules describing what we have to do if we want to fulfill the Open Source Licenses. You should know that you have to go back to the license itself.

...

3.2 Open Source and its history: some hints

Here we present main lines of the Open Source genesis: The start with the bundling of hardware and software in the beginning on the one side and the monopol of AT&T and the free distribution of unix in the universities on the other side - which together established the free hacker culture. We will shortly describe the increase of the value of software evoked by the IBM unbundling strategy and the antitrust suit against AT&T which let become the software a value itself worthful of protection and which destroyed the free exchange within the early hacker community. Naturally we will illuminate the answer of RMS, the GNU project, the founding of the FSF and the GPL. Then we will highlight the introduction of the concept Open Source invented for dissolving the troubles to talk about Free Software with managers of companies. We will hint to the Linux kernel as an unwelcome completion of the GNU system. Finally we will outline the convergency of business and Open Source, not only by Netscape/Mozilla, IBM apache, Redhat, SUN/OpenOffice but also by IBM/eclipse, Sun/Java and so on. And naturally we will highlight the meaning of 'the Cathedral and the Bazar', which had not

3 Open Source: Idea and Concepts

been written to contrast the working style of the Open Source Developemt and the proprietary 'in company' development by for example microsoft, but for discern the working and leading style of RMS and Linus.

...

3.3 Free Software versus Open Source: some hints

Here we will illuminate the differences between the ideas of the Free Software Movement and the Open Source Movement.

...

4 The Same Idea, Different Models of License

In this chapter we describe different license models which fulfill the common idea. We discuss existing types of grouping single Open Source Licenses. We highlight the limits of building such clusters for being able to analyse prototypic licenses. But finally for learning the field we ourselves cluster these license models in three groups, the Minimal Prescribing License models for protecting the developer like MIT, BSD or Apache, the Reflexive Prescribing License models for protecting the licensed code like LGPL, EPL? or EUPL and the 'viral' Overlapping Prescribing License models for protecting the on-top-developments. At the end you will know the main obligations in generally.

[TDB ...]

Be careful: Linux Magazine (04/05 2011) uses another taxonomy which seems not to be as elaborated as this taxonomy. This must be discussed[...] ...

4.1 Minimal Prescribing License models: Protecting the developer

Here we discuss licenses which only try to protect the developer, like the licenses BSD, MIT and Apache.

...

4.2 Reflexive Prescribing License models: Protecting the code

Here we discuss licenses which try to protect the developer and to protect the licensed code, like the licenses EPL?, EUPL? and LGPL

...

4.3 Overlapping Prescribing License models: Protecting the on-top-developments

Here we discuss licenses which try to protect the developer, which try to protect the licensed code, and which try to protect the on-top-developments, also known as derived works. You may expect the licenses GPL-2, GPL-3, and AGPL...

...

4.3.1 Excursion: GPL is not the Evil

Here we simply want to underline that licenses protecting the on-top-developments does not 'infect' other proprietary software automatically. It's not an aspect of the code of the license, but of the acts of the combining developers or managers. It's their decision if they have to fulfill the GPL.

...

4.3.2 Excursion: The Peculiarity of GPL-3

In the same sense we want to explain how the GPL-3 only upholds former ideas by transposing them to new technical trends.

...

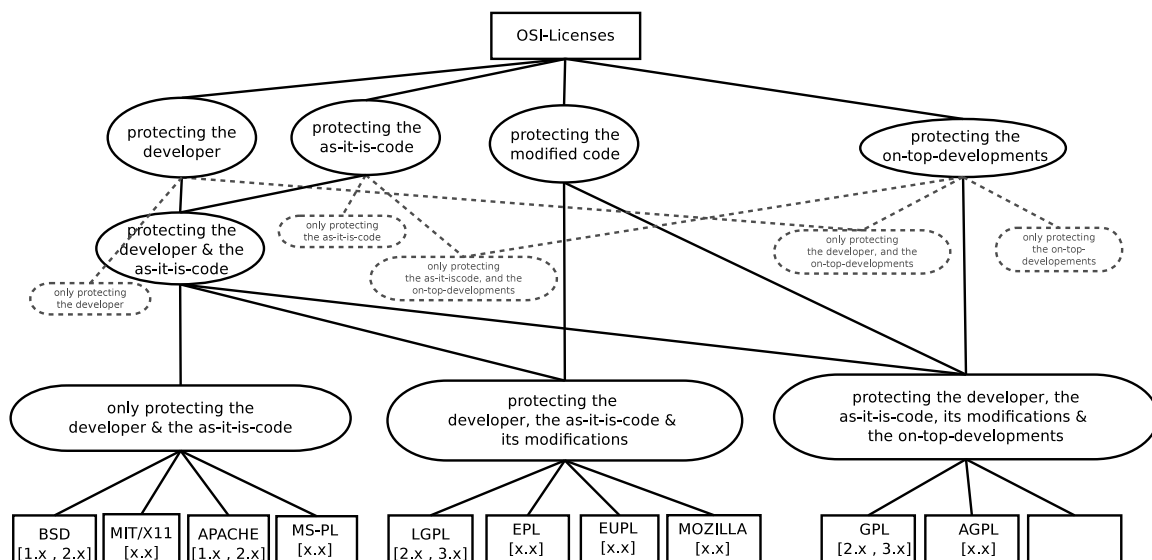
4.4 Other Models

...Do they exist? ...

4.5 Summary: Our Taxonomy for the analyzed Licenses

One pictures and the hint, that the criteria can be implemented differently by different licenses and that one therefore has to analyze the single licenses nevertheless.

...



5 Open Source: Important Minor Points

This chapter we shortly discusses some minor but although important issues.

[TDB ...]

5.1 Excursion: The Problem of Derivated Works

We will shortly discuss exsting 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's only a question of a function, method or anything else which calls (jumps into) a piece of code which has been licenced by a license protecting on-top-developements and you have a derivated work.

...

5.2 Excursion: The Problem of Combining Different Licensemodels

Here we discuss the often neglected or only loosely touched problem of combining differently licended 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 question can simply be answered: only if you embed two libraries which both are licensed by an on on-top-develeopment protecting license and if these both license require the licensing of the derivated work by diffent licenses then you have a problem. In all other case which we will describe an list there is no problem.

...

5.3 Excursion: Open Source Software and Money

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

...

5.4 Excursion: The Problem of Implicitly Freeing Patents

Here we analyze the implicitly freeing of patents by licensing code as Open Source Software.

...

6 Open Source: Use Cases As Principle of Classification

This chapter offers our taxonomie of use cases which lateron will structure each license specific chapter.

[TDB ...]

7 Open Source Licenses: Find Your Specific To-do Lists

This chapter shall offer the finder in form of a process / deciding chart based on leading questions. The main part should be a simple picture!

[TDB ...]

8 Open Source License Fulfillment: Classified To-do Lists for ...

This chapter lists for the main Open Source Licenses and for the main use cases what one has to do for fulfilling a specific license. And it explains in which cases the combination with another Open Source License must be avoided. You should be able to jump into the license specific chapter and find all relevant information - though without proving details,

[TDB ...]

8.1 Apache Licensed Software in the usage context of ...

8.1.1 Without Redistribution and ...

8.1.1.1 As-It-Is-Usage

...

8.1.1.2 With-Code-Modification

...

8.1.1.3 As-Isolated-Initial-Development

...

8.1.1.4 As-On-Top-Development

...

8.1.2 With Redistribution and ...

8.1.2.1 As-It-Is-Usage

...

8.1.2.2 With-Code-Modification

...

8.1.2.3 As-Isolated-Initial-Development

...

8.1.2.4 As-On-Top-Development

...

8.2 etc.

...

8.3 BSD Licensed Software in the usage context of ...

8.3.1 Without Redistribution and ...

8.3.1.1 As-It-Is-Usage

...

8.3.1.2 With-Code-Modification

...

8.3.1.3 As-Isolated-Initial-Development

...

8.3.1.4 As-On-Top-Development

...

8.3.2 With Redistribution and ...

8.3.2.1 As-It-Is-Usage

...

8.3.2.2 With-Code-Modification

...

8.3.2.3 As-Isolated-Initial-Development

...

8.3.2.4 As-On-Top-Development

...

8.4 GPL V2 Licensed Software in the usage context of ...

8.4.1 Without Redistribution and ...

8.4.1.1 As-It-Is-Usage

...

8.4.1.2 With-Code-Modification

...

8.4.1.3 As-Isolated-Initial-Development

...

8.4.1.4 As-On-Top-Development

...

8.4.2 With Redistribution and ...

8.4.2.1 As-It-Is-Usage

...

8.4.2.2 With-Code-Modification

...

8.4.2.3 As-Isolated-Initial-Development

...

8.4.2.4 As-On-Top-Development

...

8.5 etc.

8.6 MIT Licensed Software in the usage context of ...

8.6.1 Without Redistribution and ...

8.6.1.1 As-It-Is-Usage

...

8.6.1.2 With-Code-Modification

...

8.6.1.3 As-Isolated-Initial-Development

...

8.6.1.4 As-On-Top-Development

...

8.6.2 With Redistribution and ...

8.6.2.1 As-It-Is-Usage

...

8.6.2.2 With-Code-Modification

...

8.6.2.3 As-Isolated-Initial-Development

...

8.6.2.4 As-On-Top-Development

...

9 Open Source Licenses and Their Legal Environments

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.

[TDB ...]

10 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.

[TDB ...]

11 Appendices

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.

[TDB ...]

11.1 Some Widespread Open Source Myths

Open Source tries to improve the world ethically :- no, there's a clear ban to exclude persons, groups, purposes

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 modified (L)GPL licensed application as binary to anybody 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 person or only to all members of that group.

Published Open Source Software is open for ever :- No. The Copyright holder ever holds the Copyright. The can change the licence of next release of its software.

Software can either be Open Source Software or proprietary software :- The Copyright holders can ever distribute the code under other conditions, even additionally. That's not a question of the licence, but of the Copyright.

The opposite of Open Source Software is commercial Software :- No. Firstly 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'. Secondly there are many other forms like Freeware, Public Domain Software or anything else which is neither Open Source Software nor Commercial Software. It's senseless to take the question of money as mark 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' [Achtung: sollte eigentlich nur für GPL gelten]. 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) containing GNU software and the Linux kernel. But Ian Murdock states also, that debian does not want to earn money. (clear details)

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's 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's not required as a constitutive criteria by the OSD. It might be, that a license additionally requires your name. But's 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 responsibilities 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 do 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.

Periodicals, Shortcuts, and Overlapping Abbreviations

BISE	Business & Information Systems Engineering [ISSN: 1867-0202]
.....	Berkeley Technology Law Journal [ISSN:]
.....	Cultural Anthropology [ISSN: 1548-1360]
CiHB	Computers in Human Behavior [ISSN: 0747-5632]
CotACM	Communications of the ACM [ISSN: 0001-0782]
CRi	Computer Law Review international [ISSN: 1610-7608]
.....	Computers & Education [ISSN: 0360-1315]
.....	Cutter IT Journal [ISSN: 1048-5600]
DDT	Drug Discovery Today [ISSN: 1359-6446]
DSS	Decision Support Systems [ISSN: 0167-9236]
.....	Ethics and Information Technology [ISSN: 1388-1957]
E.C.L.R.	European Competition Law Review [ISSN:]
EER	European Economic Review [ISSN: 0014-2921]
.....	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]
.....	IEEE Software [ISSN: 0740-7459]
ifross	Institut für Rechtsfragen der Freien und Open Source Software
.....	International Information and Library Review [ISSN: 1057-2317]
.....	International Journal of Medical Informatics [ISSN: 1386-5056]
.....	interactions[ISSN: 1072-5520]
ISJ	Information Systems Journal [ISSN: 1365-2575]
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]
.....	Journal of Academic Librarianship [ISSN: 0099-1333]
.....	Journal of Comparative Economics [ISSN: 0147-5967]
.....	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'
LJ	Linux Journal [ISSN: 1075-3583]
.....	netWorker [ISSN: 1091-3556]
.....	Organization Science [ISSN: 1047-7039]
.....	Queue [ISSN: 1542-7730]
.....	R&D Management [ISSN: 1467-9310]

11 Appendices

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]
.....	Stanford Law Review [ISSN: 00389765]
.....	Software Quality 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]
.....	University of Chicago Law Review [ISSN:]
.....	University of Illinois Law Review [ISSN:]
.....	University of Pittsburgh Law Review [ISSN:]
wp.	webpage / webdocument without any internal (page)numbering

Bibliography

- 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
Originaltext der Apache-Lizenz. Beschreibt die 'Does and Don'ts'.
- Arlt, Brinkel, a. Volkmann; Spindler, Gerald, editor: 'BSD' - und 'Mozilla'-artige Lizenzen; In *Spindler*: Rechtsfragen bei Open Source Software, 2004, pp. 317–372, Print
Dies Kapitel beschreibt den juristischen Gedankengang der BSD Lizenz und der Mozilla-Lizenzen, wobei letztere als Copyleft-Lizenzen mit anderem Ansatz als die GPL dargestellt werden.
- 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 [... u.a.O.]: Waxmann, 2008, Print, ISBN 978–3–8309–1845–5
Konzentriert sich auf Patente und Open Source als Herausforderung für innovative Universitäten und bietet Artikel zu verschiedenen Aspekten dieses Kontextes
- Behlendorf, Brian: How Open Source Can Still Save the World; Konferenzbeitrag; In *Boldyreff et al.*: Open Source Ecosystems, 2009, p. 2, BibWeb/PDF
Keynote. Behauptet, dass Open Source mit ihrer Art der Kooperation insbesondere auch eine Kooperation bei der Krisenbewältigung unterstütze.
- 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 u. New York: Springer, 2009, BibWeb/PDF, ISBN 978–3–642–02031–5
Proceedings. Relevante Artikel sind einzeln aufgeschlüsselt.
- 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
Stellt wesentliche Aspekte einer Open Source Veröffentlichung gründlich, knapp und übersichtlich dar: das nicht übertragbare Urheberrecht wird gegen das transferierbare Verwertungs- und Nutzungsrecht ebenso abgegrenzt, wie Haftungsfragen diskutiert und die Wahl der Lizenz auf den intendierten Zweck hin beleuchtet werden. Trotzdem kann (und will) der Artikel kein allgemeines 'Open Source Compendium' sein: der Frage, ob und wann eine universitäre Entwicklung gezwungenermaßen veröffentlicht werden muss und was Universitäten ansonsten tun müssen, wenn sie OS Software intern nutzen und weitergeben, wird nicht beleuchtet.
- Brügge, Bernd *et al.*: Open-Source Software. Eine ökonomische und technische Analyse; Berlin u. Heidelberg: Springer, 2004, Print, ISBN 3–540–20366–4
Möchte eigenem Bekunden nach 'mit den wesentlichen Einflussfaktoren' von OSS vertraut machen: es clustert die Lizenzen und liefert neben Beispielen und Aussagen zum Entwicklungsprozess auch Aussagen über die Beteiligung von Firmen an der Nutzung und Entwicklung von Open Source Software. Interessanterweise wird aber die Pflicht, die Firmen mit der Nutzung eingehen, nicht wirklich thematisiert.
- Buchtala, Rouven: Determinanten der Open Source Software-Lizenzwahl; Eine spieltheoretische Analyse; Frankfurt am Main, Berlin, Bern [... u.a.O.]: Peter Lang, 2007 (= Infor-

Bibliography

- mationsmanagement und strategische Unternehmensführung), [Vol./No.] 12), Print, ISBN 978-3-631-57114-9
- Die Arbeit will ermitteln, welche Faktoren die Wahl einer OS-Lizenz beeinflussen. Dazu wird ein spieltheoretischer Ansatz als Kriterium gewählt. Bestehende OS-Lizenzen klassifiziert der Autor als 'permissiv', 'restriktiv' und 'hochrestriktiv'. Diese Unterscheidung erhellt inhaltlich die Bedingungen der Lizenzerfüllung. Die Namen der Klassen denkt der Autor allerdings aus Sicht einer Firma, die sich möglichst uneingeschränkt bedienen will. Dabei entfällt jedoch die Intention der OS-Lizenzgeber, gewährte Freiheiten abzusichern.*
- 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
- 'Nur' ein witziges kleines Comic-Buch, das mit der ähnlichen Aussprache der Wörter 'GNU' und 'nu' oder 'new' spielt und so den 'Revegnew' oder den 'Sgnuker' illustriert.*
- 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 ist eine beliebte Abkürzung: sie steht nicht nur für 'Open Source Software', sondern auch für 'Open Source Stacks' oder gar für 'Operation Support Systems'.*
- Coar, Ken a. Rich Bowen: Apache Kochbuch; deutsche Übersetzung v. Jochen Wiedmann; Beijing [... u.a.O.]: O'Reilly, 2004, Print, ISBN 3-89721-371-0
- Interessant insofern, als die 'Apache License', die den vielfältigen Einsatz dieses http-Servers erst ermöglicht hat, in dem Buch nicht vorkommt. Die Frage, ob und wann man diese Software nutzen und installieren darf, ob und wann welche Module verwendet werden dürfen, wird nicht gestellt, geschweige denn beantwortet.*
- De Nicolò, Christopher: Open Source Software - Rechtliche Aspekte nach deutschem und italienischem Recht. Eine rechtsvergleichende Studie. Dissertation; Regensburg: Universität Regensburg, 2010, Print
- Untersuchung, in wie weit Open Source Software - trotz europäischer Normierung - im italienischen und deutschen Urheberrecht, Vertragsrecht und Haftungsrecht je unterschiedlich behandelt wird. Bzgl. der Urheberrechte bestehe ein gleichwertiger Schutz, die 'vertragstypologische Einordnung' stelle dagegen eine Herausforderung dar. In Deutschland müsse man bei Copyleft-Lizenzen von einem Tauschvertrag ausgehen, für die anderen OS-Lizenzen von einer Schenkung. Letzteres sei in Italien aber rechtlich problematisch. In der Haftungsfrage herrsche wieder Gleichwertigkeit.*
- 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
- Beleuchtet in mehreren Artikeln das Urheberrecht im Internet. Ist nicht direkt auf das Thema 'Open Source' ausgerichtet, vielmehr auf die 'Vervielfältigung elektronischer Güter'. Behandelt aber auch die Themen '(Open Source) Software', 'Freie Kulturen' und 'Creative Commons'.*
- Drossou, Olga, Stefan Krempel, 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, Krempel, a. Poltmann: Die wunderbare Wissensvermehrung, 2006, pp. 1-10, Print
- Konstatiert, dass 'kollaborativ gefördertes Wissen frei bleiben müsse, wenn es als Anreiz für weitere Wissensvermehrung dienen solle'. Als Initialbeispiel dient der Open Source Entwicklungsprozess, der schon auf eine veränderte Erstellung von Dokumenten (GNU Free Documentation License und die Creative Commons-Initiative) u.Ä.m. hingewirkt hat. Patente stünden dem entgegen. Aber - so die Behauptung - Innovation brauche 'ein freies Spiel der Kräfte und einen möglichst breiten Wissensaustausch'.*

Bibliography

- 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*
Eine Sammlung von Artikeln, die emphatisch den freien Gedankenaustausch als notwendige Bedingung für Innovationen propagieren.
- 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*
Versucht zu zeigen, dass die Idee des 'Geistigen Eigentums' von Anbeginn an politisch umstritten gewesen ist und das Open Source die aktuelle 'State-of-the-Art'-Position daher zurecht herausfordert. Das Buch enthält eine kurze, aber aussagekräftige Zusammenfassung der 'Open-Source'-Geschichte.
- Euler, Ellen: Creative Commons: Mehr Innovation durch die Öffnung des Urheberrechts? In Drossou, Krempl, a. Poltmann: Die wunderbare Wissensvermehrung, 2006, pp.147-158, Print*
Stellt zunächst klar, dass das Urheberrecht das Trittbrettfahrerproblem lösen und so das Interesse der Allgemeinheit an neuen Ideen schützen solle. Das Recht des Urhebers, Vervielfältigungen erlauben oder unterbinden zu können, fungiere als Anreiz. Begrenzt werde dieses private Recht durch das berechnete Interesse dritter. Darauf stütze sich die Zitaterlaubnis und die Privatkopie. Im digitalen Zeitalter würden mittels DRM etc. bisher urheberrechtsfreie Nutzungen unterbunden und das Urheberrecht unterlaufen, allerdings nicht zum Schutze der Urheber, sondern zum Schutz der Verwerter. Creative Commons bietet dem Urheber die Möglichkeit, graduelle Freiheiten verlässlich zu gewähren.
- Feig, Michael: Einführung in GNU; München u. Wien: Carl Hanser Verlag, 1996 (= Unix easy), Print, ISBN 3-446-18311-6*
Ist auf das GNU-Projekt als eine Sammlung von einzelnen freien Tools ausgerichtet, die ein freies Unix konstituieren sollen. Die Lizenzfrage und die Philosophie werden jedoch nur sehr grob in einem Absatz abgehandelt. Dafür beleuchtet diese ältere Buch die einzelnen, meist in einer Shell ablaufenden Applikationen sehr detailliert. Es ist sozusagen ein praktisches Handbuch für die älteren 'Anwendungsprogramme von GNU'.
- 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*
Bekanntes Englisch sprachiges Standardwerk. Beschreibt in erster Linie das Doing der eigentlichen Entwicklung. Lizenzfragen werden in einem kurzen Kapitel abgehandelt. Enthält dort eine sehr saubere Abgrenzung der Begriffe Free Software, Open Source Software, FLOSS etc. Behandelt auch das Problem der Lizenzkompatibilität.
- Geese, Elmar: Innovation und freie Software; In Drossou, Krempl, a. Poltmann: Die wunderbare Wissensvermehrung, 2006, pp.77-84, Print*
Konstatiert, dass wir in einer Zeit der Wissensmonopolisierung leben, die über Patente, DRM und Gesetze etabliert werde. Mit der Pflege dieser Wissensprivatisierung stelle sich die Politik ihrem eigenen Ziel entgegen, Innovation zu fördern. Positive Gegenbeispiele für eine andere Kultur seien die Pflege von Staroffice/OpenOffice und die der Mozilla-Produkte.
- 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*
Erster Band der bis 2008 gepflegten Reihe. Sammelt Artikel zu heterogenen Themen.
- Gilroy, Bernard Michael a. Tobias Volpert: Die Funktionen eines Patentsystems und ihre Bedeutung für Unternehmensausgründungen aus Hochschulen; In Asche et al.: Open Source.*

Bibliography

- Kommerzialisierungsmöglichkeiten und Chancen für die Zusammenarbeit von Hochschulen und Unternehmen, 2008, pp. 21–39, Print
Skizziert die beiden gesellschaftlichen Funktionen eines Patentsystems, nämlich die, per befristetem Nutzungsmonopol Innovationsanreize zu schaffen, und die, per Offenlegung neue Informationen zu distribuieren. Zielt letztlich darauf, dass auch die Offenlegung von Informationen wenigstens mittelbar Innovation befördert, weil diese als Basis für einen aufbauenden Fortschritt essentiell sei.
- Grassmuck, Volker: Freie Software. Zwischen Privat- und Gemeineigentum; Themen und Materialien; Bonn: Bundeszentrale für politische Bildung, 2002, Print, ISBN 3-89331-432-6
Deutsches Standardwerk, entstanden zu einer Zeit, als Open Source noch nicht en vogue war. Obwohl Art und Ausrichtung der Lizenzen mittlerweile stark gewachsen sind, bietet das Buch immer noch einen systematischen Leitfaden.
- 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
Der Artikel skizziert kurz Chancen von hochschuleigenen Open-Source-Software-Portalen und meint u.a., dass solche die Softwareentwicklung und die Wiederverwendung der Ergebnisse unterstützen und dass Resultate - verbunden mit einem Imagegewinn - darüber besser in die Praxis übertragen werden.
- Heap, Nicholas: OSI-Referenzmodell ohne Geheimnis; [aus dem Englischen] übers. v. G & U, Flensburg; Hannover: Heise, 1994, Print, ISBN 3-88229-045-5
OSI referenziert zumeist nicht die 'Open Source Initiative', sondern das 'Open Systems Interconnection Referenzmodell'. Hier ein älteres deutschsprachiges Werk, das 'OSI' als Netzwerkschichtung erläutert.
- Hofmann, Susanne, Sven Pfeiffer, a. Urs Walter: Open Source School; Neue Synergien zwischen Schule und Kiez in Gropiusstadt. Architektur als sozialer Katalysator; Berlin, 2010, FreeWeb/PDF (URL: http://opus.kobv.de/tuberlin/volltexte/2010/2841/pdf/9783798322738_content.pdf)
Bietet trotz verführerischem Titel nichts, was zum Thema 'Open Source Software' gehören würde: es geht um eine spezielle Art der Architekturvermittlung.
- ifross: FAQ - Häufig gestellte Fragen; 2011, FreeWeb/HTML (URL: <http://www.ifross.org/faq-haeufig-gestellte-fragen>) – reference download: 2011-09-05
Diese Seite präsentiert Kernaussagen zum Thema 'Open Source', und zwar in Form einer FAQ-Liste.
- ifross: Lizenz-Center; 2011, FreeWeb/HTML (URL: <http://www.ifross.org/lizenz-center>) – reference download: 2011-09-05
Listet mehr Open-Source-Lizenzen auf als das OSI-Institut selbst. Sie werden eingeteilt in solche ohne Copyleft-Effekt, in solche mit strengem Copyleft-Effekt und in solche mit begrenztem Copyleft-Effekt, was eine der durchdachtesten Darstellung überhaupt ist.
- ifross: Ziele, Aufgaben, Geschichte; 2011, FreeWeb/HTML (URL: <http://www.ifross.org/node/16>) – reference download: 2011-09-05
Beschreibt Ziele und Geschichte des 'Instituts für Rechtsfragen der Freien und Open Source Software': es wurde 2000 als Privatinstitut gegründet und soll das Phänomen 'Freie Software' aus rechtswissenschaftlicher Sicht begleiten.
- Jaeger, Till a. Axel Metzger: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 1st edition. München: Verlag C.H. Beck, 2002, Print, ISBN 3406484026
Die 1. Ausgabe des wichtigsten deutschsprachigen Standardwerks zum Thema 'Open Source Lizenzen'; es behandelt wichtige Aspekte wie Haftung, Patente oder Marken. Und bzgl. jeder

Bibliography

- aufgelisteten Lizenz oder jedes erwähnten Lizenzclusters beschreibt es Rechte und Pflichten der Softwarenutzer - nur leider eben nicht in Form abarbeitbarer Listen und auch nicht unter Berücksichtigung der verschiedenen Usecases, wie sie die Softwareentwicklung kennzeichnen. Und trotzdem: es ist ein unumgebares Standardwerk, zumindest in seiner letzten Ausgabe von 2011.*
- Jaeger, Till a. Axel Metzger: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 2nd edition. München: Verlag C.H. Beck, 2006, Print, ISBN 3406538037
Die 2. Ausgabe des wichtigsten deutschsprachigen Standardwerks zum Thema 'Open Source Lizenzen': Wie ifross unterteilt es Lizenzen jetzt als solche mit 'strenger Copyleft Klausel', solche mit 'beschränkter Copyleft-Klausel' und solche ohne 'Copyleft-Klausel'. Neben anderen Aspekten diskutiert das Buch nun die Durchsetzung von Open Source Lizenzen und die Wechselwirkungen zum AGB Recht. Und erneut beschreibt es bzgl. der wichtigsten Open Source Lizenzen Rechte und Pflichten der Softwarenutzer - wiederum nur leider eben nicht in Form abarbeitbarer Listen und auch nicht unter Berücksichtigung der verschiedenen Usecases, wie sie die Softwareentwicklung kennzeichnen. Allerdings bleibt es natürlich ein unumgebares Standardwerk!
- Jaeger, Till a. Axel Metzger: Open Source Software. Rechtliche Rahmenbedingungen der Freien Software; 3rd edition. München: Verlag C.H. Beck, 2011, Print
Die 3. aktuelle Ausgabe des wichtigsten deutschsprachigen Standardwerks zum Thema 'Open Source Lizenzen': Behält die 'bewährte Kapitelstruktur' bei, listet aber neue Lizenzenh auf und nebst vielem anderem awwh auf die Lizenzkombination ein. Ergo findet man erneut zu Open Source Lizenzen die Rechte und Pflichten der Softwarenutzer - wiederum nur leider eben nicht in Form abarbeitbarer Listen und auch nicht unter Berücksichtigung der verschiedenen Usecases, wie sie die Softwareentwicklung kennzeichnen. Das scheidet der Qualität dieses unumgebaren Standardwerks nicht!
- 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
Beschreibt sehr gedrängt das Wesen von 'Wikipedia', seine Verbindung zu den Enzyklopädisten um Diderot und die Ähnlichkeit seines Entwicklungsprozesses mit dem von Open Source.
- Kersken, Sasche: Apache 2.2. Das umfassende Handbuch; 3., aktualisierte u. erw. Aufl.; Bonn: Galileo Press, 2009, Print, ISBN 978-8362-1325-7
900+x Seiten starkes Handbuch mit einem 2 seitigen Abschnitt zu Geschichte und Inhalt der Apache Lizenz.
- 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
Sehr gründliche Analyse der Gültigkeit und des Umfangs der GPL bezogen auf das allgemeine deutsche Recht, das Urheberrecht und das Vertragsrecht. Behandelt zudem detailliert und facettenreich die einzelnen GPL Bestimmung. Andere Open Source Lizenzen werden jedoch nicht in derselben Art behandelt.
- Kreutzer, Till: Software und Spiele kopieren[:] Das Lizenzmodell entscheidet; In Djordjevic et al.: Urheberrecht im Alltag, 2008, pp. 29–33, Print
Verweist auf das präjudizierende Urheberrecht Deutschlands, demzufolge bei Software Sicherungskopien zwar nicht untersagt werden können, andere Arten des Kopierens aber extra erlaubt werden müssen.
- Kreutzer, Till: Software veröffentlichen[:] Wem gehören die Rechte? In Djordjevic et al.: Urheberrecht im Alltag, 2008, pp. 163–167, Print

Bibliography

- Erläutert das initiale Zusammenfallen von Urheberrecht und Verwertungsbefugnis. Letzteres kann übertragen werden, auch generell per Arbeitsrecht: angestellte Programmierer übergeben die Nutzungsrechte an ihren Werken mit dem Arbeitsvertrag an ihren Arbeitgeber. Wichtig aber auch: Urheberrecht impliziert keinen Ideenschutz.*
- Kreutzer, Till: Softwarelizenzen - Beispiele[:] Und welche Lizenz nehme ich jetzt? In Djordjevic et al.: Urheberrecht im Alltag, 2008, pp. 176–179, Print
- Analysiert am Entwicklerinteresse entlang dazu passende Lizenzformen: Open Source Lizenzen können demnach verwendet werden, wenn keine Kontrollinteressen vorliegen, wenn Interesse am Rücklauf von Weiterentwicklungen bestehen oder wenn Werbezwecke im Vordergrund stehen; ansonsten werden proprietäre Lizenzen empfohlen. Eine feinere Unterteilung der Open Source Lizenzen wird nicht offeriert.*
- Kugler, Petra: Coordinating Innovation: Evidence from Open Source Development; Dissertation; St. Gallen: University of St. Gallen, 2005, Print
- Behandelt wird die Frage, wie sich informelle Organisationen samt ihrer Hierarchien etablieren. Open Source Communities werden als Beispiele genommen. Grob gesagt etablieren sich ihrer Strukturen auf der Basis von Themen- und Projektwissen. Open Source Lizenzen werden als Methode genommen, die den freien Austausch von Code erst ermöglichen. Wie sie konkret erfüllt werden, ist nicht Thema des Buches.*
- 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
- Untersucht den Wandel hin zur Offenheit in der Softwareentwicklung über Interviews innerhalb von 'Embedded-Linux'-Firmen. Das Ergebnis ist, dass die Offenheit kundengetrieben sei und zugleich einen Lernprozess innerhalb der Firmen bedinge. Das Buch enthält eine kurze, aber präzise Darstellung dessen, was OSS auch aus Lizenzsicht ist; eine systematische Darstellung der eingegangenen Verpflichtungen liefert es nicht.*
- Laroque, Christoph, Andre Döring, a. Thorsten Timm: 'Give or Let Buy': Kritische Überlegungen eines Software-Ingenieurs 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
- Will Aspekte der Veröffentlichung von Software als Open Source wenigstens 'schemenhaft' beleuchten. Damit hat der Artikel Erfolg. Auch Lizenz bezogene Herausforderungen werden nicht wirklich erwähnt.*
- Maaß, Christian: Zur Bedeutung des Urheber- und Patentrechts 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
- Skizziert, dass das Urheberrecht das konkrete Werk schützt, das Patent dagegen die 'dahinter stehende Idee'. Für Software sei das in Europa möglich, sofern der Hauptanspruch technischer Natur ist. Der Idee von Open Source stehe das aber entgegen. Zwecks Abwehr solcher Patent basierten Einschränkungen im OS Bereich verweist der Artikel zum Schluss auf den Patentpool, der vom OSDL aufgebaut wird.*
- Moody, Glyn: Die Software-Rebellen. Die Erfolgsstory von Linus Torvalds und Linux; aus dem Amerikanischen übers. v. Annemarie Pumpering; Amerikanische Ausgabe von aus dem Jahr 2000; Landsberg am Lech: verlag moderne industrie, 2001, Print, ISBN 3–478–38730–2
- Wenn jemand nur ein Buch über 'Open Source' lesen möchte, geben Sie ihm dies: es erzählt - von nahezu allen wichtigen Aspekten.*
- Mundhenke, Jens: Wettbewerbswirkungen von Open-Source-Software und offenen Standards auf Softwaremärkten; Berlin, Heidelberg u. New York: Springer, 2007 (= Kieler Studien, [Vol./No.] 338), Print, ISBN 978–540–71415–6
- Fragt, warum und wie Open Source Software erfolgreich im Markt agiert. Eine Antwort sei, dass OSS den Wettbewerbsdruck durch die bloße Existenz einer Alternative erhöht. Zur*

Bibliography

- Analyse der Frage folgt das Buch der 'natürlichen' thematischen Struktur: es erläutert das Konzept Open Source, klassiziert die Lizenzen in Copyleft-, Noncopyleft-, Public Domain- und proprietäre Lizenzen und skizziert die Geschichte von Open Source.*
- 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
- Monatlich veröffentliche Statistik, die Webpräsenzen und die auslieferenden Webserver zählt. Demnach ist der Apache-Webserver seit 1996 die dafür am meisten genutzte Software. Und aktuell wird sie mehr als doppelt so oft verwendet, wie alle anderen zusammen.*
- 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
- Zielt auf die Klärung der Haftung bei der Weitergabe von GPL Software. Das bedarf der Erörterung, da der Gewährleistungsausschluss in der GPL in Deutschland nichtig ist. Damit greift ersatzweise normales Recht, hier das der Schenkung. Der privaten Distributor oder Urheber haftet demnach nur bei vorsätzlichem oder grob fahrlässigen Verhalten, z.B. bei arglistigem Verschweigen von Schadroutinen. Der professionelle Distributor muss darüber hinaus Programme per Virens Scanner auf Schadroutinen überprüfen. Davon unabhängig zu bewerten, sind seine Zusatzleistungen. Neben AGB-haftigkeit der GPL und des Vertragstypus wird auch detailliert die Open-Source-Begrifflichkeit verhandelt.*
- Osterloh, Margit, Sandra Rota, a. Roger Lüthi: 'Collective Invention' als neues Innovationsmodell; In Drossou, Krempel, a. Poltmann: Die wunderbare Wissensvermehrung, 2006, pp. 65–76, Print
- Zeigt, dass 'Collective Invention' schon im 19. Jh. existiert. Gemeint ist jeweils der freie Ideenaustausch. Bisher sei er in einer vorproduktiven volatilen Phase gepflegt und mit der Entdeckung einer konstanteren Architektur beendet worden. Open Source Software Entwicklung trotz diesem Muster des Niedergangs: Das sei bedingt durch die anhaltend niedrigen Kosten der Partizipation und durch die Lizenzen, die eine Rückgabe von Innovationen an die Community erzwingen.*
- Peters, Stormy: Open Source Is Changing the Way Work Gets Done; Konferenzbeitrag; In Boldyreff et al.: Open Source Ecosystems, 2009, p. 1, BibWeb/PDF
- Keynote. Behauptet, dass die neuen Wege der Kollaboration und Kommunikation, wie sie bei Open Source Entwicklung üblich sind, auch die Zusammenarbeit von Individuen und Firmen im Allgemeinen verändern.*
- Piller, Frank T.: User Innovation: der Kunde kann's besser; In Drossou, Krempel, a. Poltmann: Die wunderbare Wissensvermehrung, 2006, pp. 85–97, Print
- Beschreibt u.a., wie bei der Entwicklung des Kite-Surfens User - sozusagen im Stile von Open Source Entwicklung - CAD-Modelle von Segeln frei erhalten und ihre Verbesserungen in Form von 'Bug-Reports' zurückgeben.*
- 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
- Diskutiert Einsatzpotentiale von OSS und offeriert eine Methodik, die Wirtschaftlichkeit ihres Einsatzes zu berechnen. Liefert eine gute Begriffssystematik und einen Softwareüberblick. Die Berechnung der Wirtschaftlichkeit beschränkt sich bei den Client-Rechnern auf eine Migration von MS-Office nach Open-Office. Pikant sind zudem die auf Interviews beruhenden Einschätzungen zu Vor- und Nachteilen von OSS. Auffallend ist auch, dass die Lizenzerfüllung nicht thematisiert wird, obwohl hervorgehoben wird, dass OSS gerade nicht frei von Lizenzbedingungen sei.*
- Rose, Marshall T.: The Open Book, A Practical Perspective on OSI; Englewood Cliffs NJ: Prentice Hall, 1990, Print, ISBN 0-13-643016-3

Bibliography

- OSI referenziert zumeist nicht die 'Open Source Initiative', sondern das 'Open Systems Interconnection Referenzmodell'. Hier ein älteres Englisch-sprachiges Werk, das 'OSI' als Netzwerkschichtung erläutert.*
- Sebal, Gerd: Offene Wissensökonomie. Analysen zur Wissenssoziologie der Free/Open Source Softwareentwicklung; Dissertation; Wiesbaden: VS Verlag für Sozialwissenschaften, 2008, Print u. BibWeb/PDF, ISBN 978-3-531-15705-4
Vom praktischen Standpunkt aus ein etwas abgehobenes soziologisches Buch: Es versucht die Bedingungen zur Möglichkeit zu ermitteln, dass sich eine offene Wissensökonomie in einer kapitalistischen Umgebung etablieren kann. OS scheint auf GPL reduziert zu sein. Lizenzfragen werden auf 3 Seiten verhandelt. Allerdings skizziert das Buch die Genese der Idee von 'Freier Software' erfolgreich. Und es unterstreicht, dass die Vorläuferform der GPL - die emacs-Lizenz - noch gefordert hatte, alle Veränderungen öffentlich zu machen, also auch die privatesten, wohingegen die GPL diesen Anspruch später aufgegeben hat.
- 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
Dient als Einleitung in die Sammlung. Der Artikel führt dazu sehr knapp in Geschichte und Systematik von Open Source ein, wobei er - der notwendigen Verknappung wegen - gelegentlich in die Irre führt.
- Spielkamp, Mathias: Creative Commons[:] Andere Zeiten, andere Lizenzen; In *Djordjevic et al.*: Urheberrecht im Alltag, 2008, pp. 219–221, Print
Sehr kurze Abhandlung über die Nutzung von Creative Commons Lizenzen
- Spielkamp, Mathias: Lessigletters-Remix[:] Die Creative-Commons-Initiative; In *Djordjevic et al.*: Urheberrecht im Alltag, 2008, pp. 223–230, Print
Beschreibt Geschichte und Ansinnen der Creative Commons Lizenzbewegung als modifizierte Übertragung des 'Gegenseitigkeitsprinzips' aus der Welt der FSF/GPL Software auf die Welt der Texte.
- Spindler, Gerald; Spindler, Gerald, editor: Rechtsfragen bei Open Source Software; Köln: Verlag Dr. Otto Schmidt KG, 2004, Print, ISBN 3-504-56080-0
Basiert auf einem Rechtsgutachten. Die meisten Kapitel stammen von Spindler selbst und behandeln die Typen von Open Source Lizenzen, das Urheberrecht, das Vertragsrecht und das Haftungsrecht. Am Ende wird zudem ein Exkurs zum Thema BSD- und Mozilla-Lizenzen angeboten. In allen Fällen wird juristisch detailliert analysiert und diskutiert.
- 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
Behauptet, dass der Begriff 'Open Source' darauf ausgelegt sei, das Recht auf freie Software nicht so in den Vordergrund zu heben. Unterstreicht dann, dass das GNU Projekt beim Begriff 'Freie Software' bleibe, um Idee und Anspruch der Freiheit ohne Ängste zu propagieren.
- 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
Erläutert die Umsetzung von 'DRM' und 'trusted computing' als eine Verschlüsselung von Daten, die werksseitig immer schon in Software und Computer eingebaut sei. Da aber in diesem Fall der Schlüssel dem Besitzer vorenthalten werde, kontrolliere letztlich das Programm respektive die Maschine den Besitzer, nicht der Besitzer seine Maschine. Und eben deshalb spricht RMS vom 'Digital Restriction Management' und vom '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

Bibliography

- Stallman betont zunächst den Unterschied zwischen Patent und Urheberrecht an einem Program: ersteres 'sichere' die Idee, letztere nur die Form. Dann diskutiert er Möglichkeiten, Patenten 'entgegen' zu treten.*
- Stallman, Richard M.: Free Software Definition; originally written in 1996; In Stallman: Free Software, Free Society: Selected Essays, 2002, pp. 41–43, Print*
Definiert die vier konstitutiven Eigenschaften Freier Software: die Freiheit, solche Programme ohne Einschränkungen zu nutzen, ihr Funktionieren zu studieren, Kopien davon weiterzugeben, und schließlich die Freiheit, solche Programme zu verbessern und diese Verbesserungen weiterzugeben. Zudem erläutert der Artikel Implikationen: Etwa mit der These, dass die Zugänglichkeit des Sourcecodes 'nur' eine Voraussetzung für diese Freiheiten sei, kein Wert an sich. Oder mit dem Hinweis, dass es also freie Software geben könne, die nicht Copyleft-Software sei.
- 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*
Eine Sammlung jener wichtigen Artikel, mit denen RMS die GNU-Philosophie etabliert hat.
- 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*
Fasst Aspekte der Idee 'Freie Software' zusammen und erzählt deren Genese aus persönlicher Sicht von 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*
Betont, dass Freie Software eine gute freie Dokumentation benötige, um praktisch nutzbar zu sein: Das Fehlen guter freier Handbücher unterlaufe die Freiheit, freie Software wirklich frei nutzen können.
- Stallman, Richard M.: The GNU Manifesto; originally written in 1984; In Stallman: Free Software, Free Society: Selected Essays, 2002, pp. 31–39, Print*
Eines der frühestes Dokumente, die die Idee des GNU Projektes und der GNU Software skizzieren. Der längste Teil des Artikels diskutiert die Vorteile der freien Software für alle Beteiligten, selbst im Hinblick auf eine kommerzielle Nutzung.
- 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*
Der Artikel skizziert Aspekte der Geschichte, der Ziele und der Philosophie des GNU Projektes.
- Stallman, Richard M.: The Right to Read; originally written in 1997; In Stallman: Free Software, Free Society: Selected Essays, 2002, pp. 73–77, Print*
Aus der antizipierten Sicht des Jahres 2096 erläutert Stallman das Dilemma eines Studenten, der seiner Freundin helfen möchte. Allerdings dürfe er das nicht, weil es dazu nötig wäre, ihr seine e-books auf seinem Rechner zum Lesen zu geben. Und genau das sei verboten und werde über eine unheilige Allianz von Verlegern und Computerherstellern technisch verhindert.
- Stallman, Richard M.: Selling Free Software; originally written in 1996; In Stallman: Free Software, Free Society: Selected Essays, 2002, pp. 63–65, Print*
Opponiert gegen den Mythos, dass mit freier Software kein Geld verdient werden dürfe. Vielmehr lege die GPL diesbzgl. gerade nicht fest, wieviel für das Kopieren der GNU Software verlangt werden dürfe: das könne kostenlos sein oder Cents, ein Euro oder Milliraden von Euros kosten.
- Stallman, Richard M.: What is Copyleft? originally written in 1996; In Stallman: Free Software, Free Society: Selected Essays, 2002, pp. 89–90, Print*

Bibliography

- Beschreibt 'Copyleft' als Methode, mit der Programme und all deren Modifikationen und Ableitungen zu freier Software gemacht werden: Zunächst unterstelle man es dem normalen Copyright. Dann erlaube man als Copyright-Inhaber die freie Nutzung, Veränderung und Weitergabe, sofern diese Erlaubnis nicht entfernt worden werde. [In Deutschland braucht der Urheber die Urheberschaft nicht über eine Copyright-Notiz zu beanspruchen und also das Gemeinfrei-Werden nicht extra zu verhindern. Als Urheber er ist immer schon der Inhaber der Urheberrechte.]*
- 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
Betont die Wichtigkeit, von 'GNU/Linux' und nicht von 'Linux' zu sprechen: Der Name 'Linux' allein erleichtere es, freie und nicht freie Software zu kombinieren, sodass am Ende die Kampagne für freie Software, weil unterlaufen, praktisch scheitern könnte. 'GNU/Linux' bewahre und transportieren als Name dagegen die Idee, freie Software auf eine spezielle Weise zu nutzen, und das selbst jenen gegenüber, die die Philosophie nicht sofort unterstützen.
- 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
Wiederholt den Standpunkt, dass die Gesellschaft Software benötige, die sie lesen, berichtigen, anpassen und verbessern könne. Stattdessen würde Eigner von Software 'Black Boxes' liefern, die man nicht untersuchen könne. Ferner erwähnt der Artikel Möglichkeiten, wie man mit freier Software Geld verdienen könne, was zudem als eine wichtige Möglichkeit herausgestellt werde.
- 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
Redet - wenig konkret - einer 'Symbiose' von offenen Strukturen und der Kommerzialisierung das Wort. Open Source scheint dabei als Ideengeber das freien Web-2.0 verstanden zu werden.
- Taubert, Niels C.: Produktive Anarchie? Netzwerke freier Softwareentwicklung; Bielefeld: transcript, 2006 (= Science Studies), Print, ISBN 3–89942–418–2
Geht als soziologisches Werk der Frage nach, in welchem Sinne die Idee von 'Open Source' resp. 'Freier Software' auch deren Erarbeitungsstil beeinflusst. Grob gesagt, muss dann die Integration heterogener Aspekte eher über Argumentationen und Kompromisse erfolgen, als durch normative Zielvorgaben.
- 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
Die zweite Hälfte des Buches führt allgemein in GNU/Linux ein, ein weiteres Viertel in DRM, Formate und verwandte kollaborative Projekte. Und das erste Viertel skizziert die Geschichte von Open Source und die Idee einiger OS Lizenzen, durchaus mit interessanten, wenn auch nicht im einzelnen belegten Details.
- Widmer, Mike J.: Open Source Software - Urheberrechtliche Aspekte freier Software; Dissertation; Bern: Stämpfli Verlag, 2003, Print
Spezifiziert sehr kompetent nebst Open Source Begriffen auch die Open Source Geschichte. Grenzt kurz 'Non Copyleft Lizenzen' und 'Copyleft Lizenzen' gegen Formen der proprietären resp. kommerziellen Software etc. ab. Analysiert sodann - von der Systematik her - die Beziehungen von Open Source Software und (schweizer) Urheberrecht und - von der Paragraphenabfolge her - die Beziehungen zwischen GPL und (schweizer) Urheberrecht.
- wikipedia (de): Free/Libre Open Source Software; n.l., 2011, FreeWeb/HTML (English Version under http://en.wikipedia.org/wiki/Free_and_open_source_software) (URL: <http://de.wikipedia.org/wiki/FLOSS>) – reference download: 2011-09-08

Bibliography

Beleuchtet den Versuch, die Trennung von Free-Software-Bewegung und Open-Source-Vereinigung zumindest konzeptionell durch die Nutzung hybrider Kunstbegriffe ala F/LOSS zu überwinden.

wikipedia (en): MIT License; n.l, 2011, FreeWeb/HTML (URL: http://en.wikipedia.org/wiki/MIT_License) – reference download: 2011-09-20

Erläutert klar, dass es mehr als eine MIT-Lizenz gibt, nämlich die 'einfachere' MIT-expat-Lizenz und die etwas komplexere MIT-X11-Lizenz. Aber im Kern gehören sie zur selben Lizenzklasse: sie schützen den Entwickler, nicht den Code.

Williams, Sam: Free as in Freedom. Richard Stallman's Crusade for Free Software; Beijing [... a.a.O.]: O'Reilly, 2002, Print, ISBN 0-596-00287-4

Die erste(?) Biographie über RMS: sie enthält so viele Interviews and Hintergrundinformationen, dass sie als Primärquelle genommen werden kann, und zwar nicht nur für die Geschichte von RMS selbst, sondern auch für die Geschichte der 'Freien Software' und ihrer internen Gegenspielerin 'Open Source'.