

EXTENDED PROJECT

OPEN SOURCE SOFTWARE

An investigation into the present state of free / libre and open source software, open source community trends and current challenges to the open source community.

September 5, 2017

John Hawkes

ID: 151083

Palmer's College

Abstract

My project aims to explore both the present and future of open source software in multiple regards:- Its usage, openness and the motivations of contributors.

In addition to exploring in greater depth the community behind these software projects, I also aim to introduce the reader to the free / libre community, and the open source development methodology that they follow to ensure a greater understanding of both the worth of open software projects, and the ideological and practical issues that they face.

The Free / Libre and open source communities are ever evolving due to their non-monolithic nature, with the entire community and sub groups in control, many changes to opinions can occur in a very short time, leaving this a very interesting and dynamic topic that is, quite literally open to investigate, and always promising a new range of issues and innovations.

Contents

1	Introduction to free, libre and open source software	3
1.1	What is free software?	3
1.2	Summary of key differences	4
2	Investigation methods	5
2.1	Reading material	5
2.2	O'Reilly OSCON London	6
2.2.1	Event Information	6
2.2.2	Attendees	6
2.3	Community Surveys	6
I	Investigation	7
3	Usage of FLOSS	8
3.1	Present	8
3.1.1	OSCON 2016	8
3.1.2	Security of Free / Open source systems	9
3.1.3	Reliability of Free / Open source systems	9
4	How open is Free, Libre and open source software?	10
4.1	Licensing	10
4.2	The threat posed by proprietary open source	11
4.2.1	MariaDB MaxScale Proxy Licensing	12
4.3	Community reactions to "proprietary shifts" in open source projects	13
4.3.1	OwnCloud / NextCloud project Split	13
5	Review and Conclusions	14
5.1	Preventing the fragmentation of the Open Source community	14

5.2	Protecting open source from business acquisition	15
5.3	How businesses can contribute fairly	15
5.4	How software respect user freedoms whilst making an income	16
Bibliography		18

Chapter 1

Introduction to free, libre and open source software

The key differences, and similarities, between libre and open source software are various, and primarily based on project ideology, rather than the program code itself. Free and libre projects are largely bound by the terms of the GNU Project, and the Free Software Foundation, which conveys a message of "Free as in freedom". This definition of software provides a true benchmark for software openness that can be further compared to the present state of open source software; a movement which the Free Software Foundation and others try to distance their selves from due to the advertising of benefits to code quality, rather than the benefits to user freedoms and the other social values of the free software movement. This difference is quite well summarised by the Free Software Foundation; Describing open source as a development methodology, and free software as a social movement.

1.1 What is free software?

As described above, and in the Free Software Foundation's definition, the basis of free software is a social movement that aims to promote user freedoms; through a key set of freedoms as described on the Free Software Foundation's website¹.

The key principles of software freedom revolve around protecting the rights of the user against a the erosion of their freedoms to control software in their use, by allowing them to run the program as they wish, and for any purposes, Freedom "0", be able to see how the program works, and change it to meet your individual requirements², Freedom 1, The

¹Free Software Foundation, *What is free software?*

²This requires access to the program's source code

freedom to be able to share the software (and your modifications) with others, Freedom 2 / 3, Which combine to allow the user of the software to fully control, modify and share the software as they see fit, which greatly opposes the current models of many proprietary software companies who impose Digital Rights Management (DRM) to control the usage of their software and ensure compliance, breaching user privacy and their ability to truly own and control software in the process.

1.2 Summary of key differences

Key differences between open source and free software are eloquently put by Stallman, who describes Open Source as a development methodology, and Free software as a social movement:- Whilst in some ways this is very true, I personally feel that in many ways open source software can be considered a movement in and of itself, offering many of the freedoms expressed by the Free Software Foundation; though, in cases where there is not complete overlap, to a more restrictive degree (from the perspective of user freedoms) - Issues arise, as discussed later in the document, when companies and projects revert to licenses which restrict user freedoms in order to exploit their creations for profit, creating quasi-open source applications which use Open Source as a badge to attract custom; giving a false impression of full software freedom. This issue is further discussed in section 4.2

Chapter 2

Investigation methods

2.1 Reading material

Due in part to the nature of the topic, there is little formal written material with regard to the open source community in general, with the exception of some non-academic books. Most literature on the topic, due to its relation to enterprise, are sponsored white papers or other surveys paid for by many of the open source companies that I discuss in the body of my work, with many other sources being written by business journalists sponsored to encourage product uptake etc. As such, to reduce the impact of the many and varied community biases, I have taken my sources from a large, wide variety of sources where written sources are needed, and avoided their reuse where possible.

Further to this, I have taken information where possible from the most relevant site, where I feel bias does not impact the justification of my own message, as in MariaDB Corporation AB (*Adopting and Developing BSL Software*) and NextCloud (*Why the (A)GPL is great for business users*), to ensure that the information I gain is not unduly tainted by the often opinionated news sources that I use elsewhere to justify my points, especially in these two issues where outside commentary is often laden with justifications for actions by the aforementioned projects, or with denunciations of their actions by proponents of user freedoms or other.

In some instances, the literature read is not directly sourced from one source, but from multiple, to prevent the impact of the aforementioned sponsorship bias that can be expected in professional publications for enterprise, which are often paid to promote various products. This use of multiple sourcing also helps to overcome the other issue noted above, regarding the opinions of the journalist in question, and their personal feelings to an event.

2.2 O'Reilly OSCON London

2.2.1 Event Information

OSCON is an event focused on all aspects of open source software and the open source community, with attendees ranging from large businesses and open source projects to foundations, such as the Software freedom conservancy. In addition to the stands on the conference floor, there were a variety of talks from external speakers and companies covering a broad variety of topics, from software openness to practical advice for the implementation of open source projects in enterprise settings.

2.2.2 Attendees

On the conference floor, I was able to meet and exchange contact details with a large variety of open source projects and open software foundations. Amongst the attendees were representatives from many groups that focus on the preservation of software openness and enforcement of open source licences.

2.3 Community Surveys

Though I had initially planned to conduct a community survey, due to the wide variety of topics and sheer size of the community, I did not feel that it was possible to create and distribute a questionnaire to a large enough segment of the community while keeping questions relevant to their special interests without biased responses. Further to this, I feel my project to be well enough supported by the varied written sources, and my visit to the O'Reilly OSCON.

Part I

Investigation

Chapter 3

Usage of FLOSS

3.1 Present

3.1.1 OSCON 2016

It was a widely held view at OSCON that over the past year, open source had "won" in enterprise applications, with a massive surge in the use of open source software in corporate and enterprise environments. Many attendees cited the progress made by the cloud in popularising open source applications and software.

Based on my personal perceptions, whilst this OSCON displayed the great progress open source has made in industry, and as such is reaping the rewards of dedicated open source development and contribution teams, it still has a great deal of progress to make in the terms of user experience and uptake in user facing applications - Open software was intended to be about people being able to give and take as an individual, not as a collective of companies, and this continued growth in an already Linux / open source dominated server and back end environment does little to motivate a spirit of encouraging openness in software on an individual level, which, in the long term, will be the key to ensuring the openness of end user products.

Amongst the many exhibitors at OSCON, it was, in some ways, surprising to see representatives from closed-source monoliths such as IBM and PayPal, and it is an indicator of the progress amongst open source communities that these large companies feel the need to exhibit to open source developers, in addition to the benefits this entails for open source, as these companies appreciate the rise of, and need to cater to open source developers and enterprise users, and hence the trickle down benefits of software openness for end users.

As could be expected of an industry based event, many of the open source projects

in attendance were not GNU Licensed, but rather based on the idea of open source as a development methodology. There was a marked difference in presentation between the free software projects and open source projects in attendance, that made the profitability of well marketed open source projects very clear.

3.1.2 Security of Free / Open source systems

It is often found when making use of free and libre systems that due to the openness of code, security through obscurity is largely eliminated, with projects having to be mindful of the fact that potential attackers can see the source code, and manipulate any bugs or weaknesses that they find without disclosure:- Whilst this may seem to make open source systems more vulnerable, as aforementioned, any poor practices in source code can be seen by the entire community, and rectified, far more quickly than in closed source software products where, whilst exploits may take longer to discover initially, they may remain unknown by the software company for many years before finally being discovered, a situation that occurs surprisingly regularly, as testers discover fresh vulnerabilities and disclose them, many of which have either been dormant, or exploited by malicious parties without knowledge.¹

3.1.3 Reliability of Free / Open source systems

Similarly to section 3.1.2, it is often found that open source software is more reliable in many instances², due in part to the participation of the entire community in development, allowing for more esoteric use cases to be well catered for, In addition to this, the flexibility of development and various skill levels and skill sets involved in contributing to and open source project helps to ensure that work is checked before it is committed to a stable version, improving the stability of released software. Furthermore, many software projects maintain a variety of automated code builds / release versions, such as nightly, daily and weekly versions to facilitate regular testing of new functionality and fixes by the entire user base, with stable builds that are supported for longer periods of time, and added to more slowly to ensure that there are no issues in newly released code.

¹It is notable that openness is one of the best ways to enforce security, even in closed source applications, through the use of responsible disclosure practices, where, should the company fail to rectify the security vulnerability in the agreed time frame, a researcher publishes the nature of the exploit and how to use it, removing any false sense of security, and ensuring that the company responsible receives a great deal of negative press coverage. This shows how effective openness is at motivating rapid change, even in closed source environments.

²Miller et al., "Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services".

Chapter 4

How open is Free, Libre and open source software?

4.1 Licensing

The licences available to Open source software projects are many and varied, with various levels of permissiveness, and conditions imposed on the developers. In recent years, as the number of licenses has increased, it is commonly a hot topic of debate amongst developers, contributors and communities, in addition to the many foundations who work to further adoption of permissive licenses, and ensure that open source licenses are enforced. GNU Enforcement activities are one of the primary areas in which the Software Freedom Conservancy play a part. A key justification for these efforts, in addition to maintaining the GNU philosophy of openness, is that many open source software projects heavily rely on development contributions from external corporate users who, without a motivation to openness and contribution, would most probably include the code in their for-profit products without any return for the project or its community. This semi-reliance on external development, sponsored by companies, is clearly demonstrated by the huge progress of Ubuntu and Open stack with the recent progression of cloud computing, which is heavily reliant on open source code, as aforementioned.

Due to the additional adoption by businesses, Open source as its own entity has greatly expanded, however, this comes at a cost to the original ideologies of the Free Software Foundation and GNU projects. With this shift in utilisation, more and more open source *companies* have begun to exist; publishing parts of their source in an appeal to community contributors, but charging for their code - Whilst this in itself does not constitute a violation

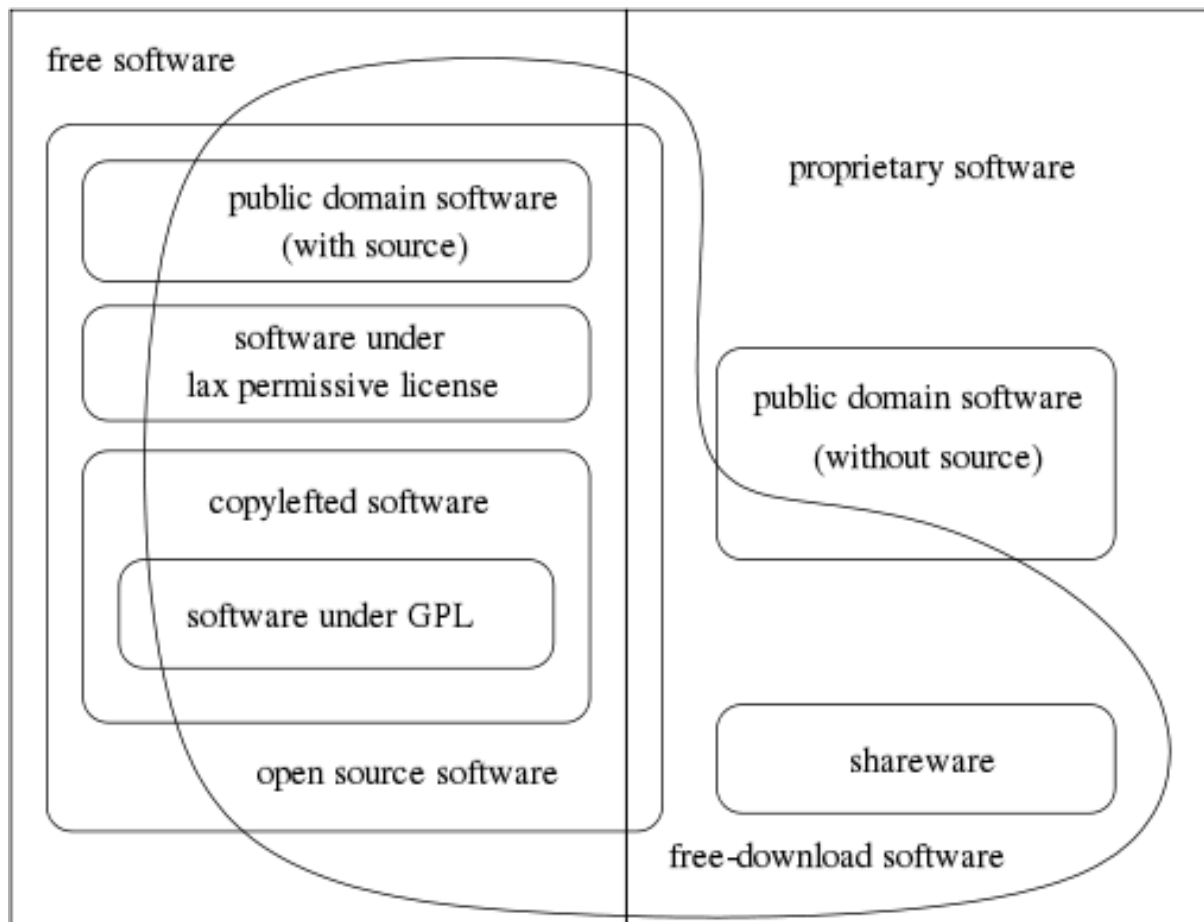


Figure 4.1: Categories of free and non free software

2

of the principles of the free software movement¹, it often means that source code is only partially available, placing the very concepts of both open source and free software - the idea that by sharing our best knowledge, and sharing alike, we can progress much more quickly.

4.2 The threat posed by proprietary open source

By referring to figure 4.1, we can see small, but noticeable area where Open source software crosses into the realm of proprietary software. As the latter half of section 4.1 begins to describe, as the opportunities in enterprise circles grow, so do those looking to exploit them:- There is far more made in enterprise software sales than in the consumer market, making the

¹A famous quote from Richard Stallman that describes this well is "Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer." This constitutes one of the cornerstone beliefs of the free software movement, that there is a difference between *free* and *gratis*

attraction of open source an excellent unique selling point for those who wish to market to business, with Open Source representing to many rapid, continuous development of features and continual maintenance. The greatest threat to free software ideals in proprietary open source is caused by projects which convert from a GPL compatible license to a proprietary, occasionally self produced software licenses that restrict user freedoms to view current versions of source code, damaging the community in the interests of profit. One of the greatest examples of a software licensing change away from GPL to a more business-restricting license is the MariaDB proxy, as in 4.2.1

4.2.1 MariaDB MaxScale Proxy Licensing

In the past year, little has attracted more attention to the dangers of licensing changes than the change implemented by the MariaDB foundation in the latest releases of their MaxScale Proxy; moving from the Free / Libre GPL Compatible licensing that they currently make use of in their other projects, to a more restrictive, self made license that disallows sharing of source code initially, instead releasing the (outdated) source code as it is obsoleted by the next version. This change was made with the intention of enabling charging for large scale enterprise use of the proxy, However, it raised concerns around the project's dedication to the Free Software movement, however, it seems to many that the project's decision was the right thing to do, as it allows further financing of the development through enterprise licensing fees:- The key argument in it's favour is found both on the MariaDB Project FAQ regarding the Business Source License³, and in the personal blog of Michael Widenius⁴, the initial developer of MySQL, and current CTO Of the MariaDB corporation, specifically the need to finance further development of the MariaDB software. This is a common dilemma for open source projects, especially those developed to be business orientated as they can often feel exploited by business users that make use of the software freely, without proving any recompense either in the form of monetary donations to support development, or by funding development teams to work on the projects; a more common form of contribution for many companies that has lead to large employment opportunities for open source developers, who are paid by companies to add desired features to open source projects, in turn contributing back to the project.

³MariaDB Corporation AB, *Adopting and Developing BSL Software*.

⁴Widenius, *Applying the Business Source Licensing (BSL)*.

4.3 Community reactions to "proprietary shifts" in open source projects

4.3.1 OwnCloud / NextCloud project Split

The OwnCloud project is an example of the risks that are somewhat exclusive to open source development; where the projects' code base was *forked* to a new project, titled NextCloud, as a result of differences in opinion among high level contributors, in a case extremely relevant to this thesis, as to the direction in which to proceed:- Other major contributors to the OwnCloud project wished to create closed source enterprise features, whilst maintaining an Open Source "Core"⁵ whereas the NextCloud project founder, once one of the largest contributors to the OwnCloud project, wished to continue development with an entirely open source, GPL compatible code base, a "feature" which they actively promote as beneficial to users, individuals and enterprises alike⁶.

It is only due to the freedom to diverge the code base that this was able to occur, and demonstrates well the key principles of software freedom, in addition to the community willingness to act to preserve these freedoms in many case, with developers supporting the new platform, and users transitioning in turn.

⁵This is mirrored in many of the other "Proprietary open source" cases in referenced across this thesis, and is especially notable in this instance for the immediate and extreme backlash caused as a result, demonstrating one of the key benefits and risks that free and open source development provides; the ability for developers and users to freely take the code base on a different course from that which is decided by a project leader, or in this case a body of major contributors, unlike in a proprietary software company, where you would expect far more rigorous enforcement of hierarchy, in addition to licensing that prevents divergence of the code base into a separate project.

⁶NextCloud, *Why the (A)GPL is great for business users*.

Chapter 5

Review and Conclusions

5.1 Preventing the fragmentation of the Open Source community

I feel that the present course of open source development is leading towards a fragmented open source community, with communities focused on business software drifting away from free software principles, whilst those developing for home use and individuals are tending towards the GPL, due either to the desire to exploit the lucrative business software market, or to ensure the continued development of their projects through securing continuing funding by charging licensing fees; hard to do with the free sharing of source code as required by GPL compatible licensing.

This is often driven by a feeling that the businesses unfairly exploit their free open source efforts, by making use of their software without contributing in any form; either intellectually or monetarily, whilst using the software to generate large amounts of income. As such, projects developed by individuals and used by businesses can shift towards closed source / non GPL Compatible licensing, as in the case of iTextPDF, One of the companies that was present at OSCON 2016.

I requested comments from their engineer regarding their move away from Open Source¹, And towards proprietary licensing for business usage and redistribution in programs, backed / distributed by the iText software corporation, in place of the usual project that would be expected for open source software.

Based on both my discussion at OSCON 2016, and the changes that the project / corpora-

¹iTextPDF was initially developed under the Mozilla Public License and LGPL, up to version 2.1.7 (Java) where the project instead moved to the AGPL v3.

tion made to their licensing policy, it was felt that the developers wished to make an income on the project as it was being used to generate income for others, with little recompense for their time or efforts:- Businesses were using *and abusing*, in the opinion of the maintainers, leading to a shift towards the closed source licences. It is relatively clear in this instance that if the users of the project were more willing to contribute back to the project as is expected in free software, the project would be far less likely to move towards a closed source licence.

5.2 Protecting open source from business acquisition

In some cases, businesses can form a more direct threat to the open source community, by purchasing the rights to a project and removing its source code, partially or entirely from the public domain, as was the case with MySQL, which was acquired by Sun Microsystems²³, which was subsequently acquired by the Oracle Corporation. Whilst, fortunately, it was, and is still maintained as a GPL open source project, this acquisition lead to a strong reaction from the open source community, With the MariaDB Project (as referenced in section 4.2.1) forking⁴, under the direction and development of the original creator Michael Widenius, in addition to many disenchanted users seeking alternatives in PostgreSQL and MongoDB.

5.3 How businesses can contribute fairly

As discussed in other parts of the review, there are two primary methods for enterprises to give back to open source projects that they make use of, both of which lead to returns on investment for said company; either providing monetary investment or hiring in professional open source developers, as has become practice in many technology companies⁵⁶.

There are a multitude of benefits for contributing companies, as monetary or intellectual contributions would both give said company and opportunity to steer the development of the software and the roadmap of future features, in the case of intellectual contributions, the

²Sun Microsystems, *Sun Microsystems Announces Completion of MySQL Acquisition; Paves Way for Secure, Open Source Platform to Power the Network Economy*.

³Tecmint, *The Story Behind Acquisition of MySQL by Sun Microsystem and the Rise of MariaDB*.

⁴Forking is one of the most powerful tools in the arsenal of the open source community; It essentially means taking a complete copy of a projects' source for further development under another banner, in doing so, this often splits development teams / contributors and users, with forks often occurring when there are inconsolable differences among large contributors / developers as to which direction the project should go in. For further reading on the fractious nature of project forks, you should see section 4.3.1

⁵Gold, *Almost two-thirds of software companies contributing to open source*.

⁶Kowalski, *Why your employees should be contributing to open source*.

ability to directly add any features that they wish to include⁷

5.4 How software respect user freedoms whilst making an income

Although encouraged by the Free Software Foundation⁸, it is a great challenge, at present, for software licensed under any free software license to generate income, with the various GPL compatible licenses being extremely permissive:- Though you can sell a compiled⁹ program for any charge, sold software must include full access to the program's source code, defined in the GPL license¹⁰ either freely, or for a fee no greater than the reasonable cost to practically supply the source code¹¹ which then leads to a larger issue - The freedom of these paying users to redistribute the software at any price, or free, severely damaging the program's marketability; as anyone could publish a free copy of the software.

This permissiveness stems from the primary user freedoms as outlined by the Free Software Foundation, which, unless the Free Software Foundation sees a significant change in ideology, remain unchangeable, As these principle software freedoms form the basis of the entire free software movement.

With this, it seems that the best method I have found for an open source software project to gain an income comes from the provision of accredited support services, which not only fully respects the FSF's user freedoms, but also eases and enables deployment and use by larger companies and enterprises that, often as a matter of company IT policy require vendor support for their software. This model of operation has been used by a number of large open source projects, most notably employed by canonical, the present maintainers of Ubuntu.¹² This gives confidence to business adopters of their software confidence, whilst also enabling them to produce an income by marketing their uniquely detailed knowledge of the operating system. In addition to the offering of support services, Canonical also offer the "Landscape"

⁷These features would be submitted to the project for approval by other users and contributors before being merged with the original source code, and introduced as a feature for all other users. By the nature of open source, if these features are not added to the original program, the business in question could continue development as a separate branch from the original source code, or an entirely separate project, merging other desired changes from the original source as needed.

⁸Fioretti, *Is it legal to sell GPL software?*

⁹compilation refers to the process where a program's source code, the easily human readable / editable part of a program is converted into uneditable, unreadable machine code that can be executed by the computer. Many users will be familiar with the ".exe" extension, which signifies an executable, compiled program.

¹⁰Subsection 6

¹¹An example would be the cost of a flash drive and postage, with no margin for profit

¹²Canonical, *Ubuntu Advantage*.

management system¹³, which follows the more common principle of a closed source module, useful to businesses which is available for a license fee. Whilst other methods could be used to complete the same tasks, this is a simple method for a Business to ease their deployment with vendor supported software.

The various methods that I have discussed above for gaining an income from the production and maintenance of free, open source software packages

¹³Canonical, *Ubuntu Landscape*.

Bibliography

- [16] GNU Affero General Public License. Free Software Foundation. 2016. URL: <https://www.gnu.org/licenses/agpl.html>.
- [And16] Tim Anderson. *Bash on Windows. Repeat, Microsoft demos Bash on Windows*. 2016. URL: http://www.theregister.co.uk/2016/03/30/bash_shell_comes_to_windows_10/.
- [Bha16] Swapnil Bhartiya. *Bash on Windows: What Does It Mean?* 2016. URL: <https://www.linux.com/news/bash-windows-what-does-it-mean>.
- [Bri16] Peter Bright. 2016. URL: <http://arstechnica.com/information-technology/2016/11/microsoft-yes-microsoft-joins-the-linux-foundation/>.
- [Can17a] Canonical. *Ubuntu Advantage*. Canonical. 2017. URL: https://buy.ubuntu.com/?_ga=1.252101263.704436347.1486631120.
- [Can17b] Canonical. *Ubuntu Landscape*. Canonical. 2017. URL: <https://landscape.canonical.com/>.
- [Cho16] Choosealicense.com. *Choosealicense.com*. 2016. URL: <http://choosealicense.com/licenses/>.
- [Cor16] Microsoft Corporation. *Looking Forward: Microsoft Support for Secure Shell (SSH)*. 2016. URL: <https://blogs.msdn.microsoft.com/powershell/2015/06/03/looking-forward-microsoft-support-for-secure-shell-ssh/>.
- [Fio13] Marco Fioretti. *Is it legal to sell GPL software?* TechRepublic. 2013. URL: <http://www.techrepublic.com/blog/linux-and-open-source/is-it-legal-to-sell-gpl-software/>.
- [Fou16] GNU Foundation. *Philosophy of the GNU Project*. 2016. URL: <https://www.gnu.org/philosophy/>.
- [Fre16a] Free Software Foundation. *GNU General Public License*. Free Software Foundation. 2016. URL: <https://www.gnu.org/licenses/gpl.html>.

- [Fre16b] Free Software Foundation. *GNU Lesser General Public License*. Free Software Foundation. 2016. URL: <https://www.gnu.org/licenses/lgpl.html>.
- [Fre16c] Free Software Foundation. *What is free software?* 2016. URL: <https://www.gnu.org/philosophy/free-sw.html>.
- [Gol16] Jon Gold. *Almost two-thirds of software companies contributing to open source*. Network World. 2016. URL: <http://www.networkworld.com/article/3062031/open-source-tools/almost-two-thirds-of-software-companies-contributing-to-open-source.html>.
- [Ham+16] Jeffrey Hammond et al. *The Tenth Annual Future of Open Source Survey*. 2016. URL: <https://www.blackducksoftware.com/2016-future-of-open-source>.
- [Kow16] Robert Kowalski. *Why your employees should be contributing to open source*. 2016. URL: <https://blog.codeship.com/why-your-employees-should-be-contributing-to-open-source/>.
- [Mar16] MariaDB Corporation AB. *Adopting and Developing BSL Software*. MariaDB Corporation AB. 2016. URL: <https://mariadb.com/bsl-faq-adopting#purpose>.
- [Mil+95] Barton P Miller et al. "Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services". In: (1995). URL: https://web.archive.org/web/20100621162832/http://ftp.cs.wisc.edu/pub/paradyn/technical_papers/fuzz-revisited.pdf.
- [Nex16] NextCloud. *Why the (A)GPL is great for business users*. NextCloud. 2016. URL: <https://nextcloud.com/blog/why-the-agpl-is-great-for-business-users/>.
- [ORe16] O'Reilly. "London OSCON 2016". In: Oct. 2016. URL: <http://conferences.oreilly.com/oscon/open-source-eu>.
- [Sun08] Sun Microsystems. *Sun Microsystems Announces Completion of MySQL Acquisition; Paves Way for Secure, Open Source Platform to Power the Network Economy*. Sun Microsystems. 2008. URL: <https://web.archive.org/web/20080228025123/http://www.sun.com/aboutsun/pr/2008-02/sunflash.20080226.1.xml>.
- [Tec15] Tecmint. *The Story Behind Acquisition of MySQL by Sun Microsystem and the Rise of MariaDB*. Tecmint. 2015. URL: <http://www.tecmint.com/the-story-behind-acquisition-of-mysql-and-the-rise-of-mariadb/>.

- [Vau16] Stephen J. Vaughan-Nichols. *Linux and Open Source Software have won, Get over it*. 2016. URL: <http://www.zdnet.com/article/linux-and-open-source-have-won-get-over-it/>.
- [Wid16] Michael Widenius. *Applying the Business Source Licensing (BSL)*. 2016. URL: <http://monty-says.blogspot.co.uk/2016/08/applying-business-source-licensing-bsl.html>.
- [Wil02] Sam Williams. *Free as in Freedom: Richard Stallman's Crusade for Free Software*. O'Reilly Media, 2002. ISBN: 0-596-00287-4.

List of Figures

4.1 Categories of free and non free software	11
--	----