

Literature review of legal cases in free and open source software, open hardware and open data

**Project ZOOM - 3Os and IP Awareness
raising for collaborative ecosystems**

Grant Agreement No. 101007385



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/). <https://creativecommons.org/licenses/by-sa/4.0/>



Funded by the
European Union

Document Control Information

Settings	Value
Document Title:	D1.1 'Literature review of legal cases in free and open source software, open hardware and open data'
Project:	'3Os and IP Awareness raising for collaborative ecosystems' (ZOOM), Grant Agreement No. 101007385
Document Version:	V17
Dissemination Level:	PU – Public
Date:	24/03/2023

Document Author(s)

Organisation	Name(s)
KUL	Ivo Emanuilov
FSFE	Lucas Lasota
FSFE	Niharika Singhal
VTT	Jutta Suksi
UP	Elizabeta Zirnstein

Table of Content

Table of Content.....	2
List of Abbreviations	7
List of Figures	9
List of Tables.....	10
1. Executive Summary	11
1.1. Project Outline	11
1.2. Purpose of the Document	12
2. Open Source as a Development and Licensing Model	13
2.1. Definition of “Openness” and “Open Source”	13
2.2. Open Source Software.....	16
2.3. Open Hardware	19
2.4. Open Data.....	23
3. Principles of Free and Open Source Software Licensing	27
3.1. Introduction	27
3.2. Copyright and Frameworks for Reuse of Software	29
3.3. Defining Free and Open Source Software	30
3.4. Types of Free Software Licenses	34
3.5. Reciprocity of Free Software Licenses.....	34
3.5.1. Non-reciprocal Licenses (Permissive).....	35
3.5.2. File-scoped Reciprocal Licenses (Weak Copyleft).....	35
3.5.3. Project-scoped Reciprocal Licenses (Strong Copyleft)	36
3.6. Other Types of Licenses in Software Projects.....	37

3.6.1.	Licenses for Documentation, Data, Image, and Video	37
3.6.2.	Licenses for Data to Be Used in AI-powered Software	38
3.6.3.	Licenses for Hardware	39
3.7.	Free Software License Compatibility	40
3.8.	Free Software Licensing and Software Patents.....	42
3.8.1.	Granting Patent Rights.....	44
3.8.2.	Retaliating in Case of Patent Aggression	44
3.9.	Free Software Licensing and Export Control Rules	47
3.9.1.	Dual-use Items and Export Control.....	48
3.9.2.	EU Export Control Regime	48
3.9.3.	Export Authorizations and Software.....	49
3.9.4.	Free Software and EU Export Control Regime.....	49
3.9.5.	Encryption Products and EU Export Control Regime.....	50
3.9.6.	Cyber-surveillance Items.....	51
3.9.7.	Compliance with EU Export Control Regime	51
3.10.	Conclusion	52
4.	Primer on Free and Open Source Software Licensing Compliance.....	54
4.1.	Fundamentals of Free Software Licensing Compliance.....	55
4.2.	Choosing a FOSS License	56
4.3.	Common Free Software Licenses	58
4.3.1.	Project-scoped Reciprocal Licenses (Strong Copyleft)	62
4.3.2.	File-scoped Reciprocal Licenses (Weak Copyleft).....	64
4.3.3.	Non-reciprocal Licenses (Permissive)	66
4.4.	Introduction to Free Software Licensing Compliance Program	67
4.5.	Standardizing License and Copyright Information for Software Projects: the REUSE Initiative.....	71

4.5.1. REUSE Specifications.....	73
4.5.2. REUSE and Related Compliance Initiatives	78
4.6. Free Software Contemporary Issues: the Right to Repair and Regulation Over Corporate Behaviour Towards Digital Devices	79
4.6.1. Right to Repair and Free Software	80
4.7. Regulating Corporate Behaviour Towards Digital Devices in the EU.....	84
4.8. Conclusion	86
5. Primer on Open Hardware Licensing.....	88
5.1. Scope of “Open Hardware” and Licensing Issues	88
5.2. Reciprocal Licenses and Open Hardware	90
5.2.1. CERN Open Hardware License	91
5.2.2. TAPR	94
5.3. Non-reciprocal (Permissive) Licenses and Open Hardware.....	96
5.3.1. CERN-OHL-P	96
5.3.2. Solderpad Hardware License	96
6. Principles and Concurrent Phenomena of Open Data Licensing.....	99
6.1. Data, Intellectual Property and Intellectual Property Rights	99
6.1.1. What Exactly is Data?	99
6.1.2. Data as Intellectual Property	100
6.1.3. Data as a Service	102
6.2. Types of Open Data Licenses	102
6.2.1. Traditional Categorization of Open Data Licenses.....	103
6.2.2. Compatibility.....	103
6.2.3. Problems of Using Traditional Licenses for Data	104
6.2.4. Novel Approaches	105
6.3. Concurrent Topics Regarding Open Data	106

6.3.1. Trend towards services and ecosystems	106
6.3.2. Regulatory Trends	107
6.3.3. Role of Data Management, Data Governance and Compliance	108
6.4. Building Business on Open Data	109
7. Definition of ZOOM Legal Cases	112
7.1. Machine Learning Programming Assistance Software	113
7.2. Quantum Programming Language	113
7.3. Industrial Robotics Device Design	114
7.4. Blockchain.....	114
8. Conclusion	115
Bibliography	116
Appendix.....	124

List of Abbreviations

Acronym	Explanation
AGPL	Affero General Public License
AI	Artificial intelligence
BSD	Berkeley Software Distribution (a Unix-like operating system)
CC	Creative Commons
CDLA	Community Data License Agreement
DMA	Digital Market Act
EC	European Commission
ECJ	European Court of Justice
EU	European Union
EUPL	European Free/Open Source Software (F/OSS) licence
FOSS	Free and open source software
FSF	Free Software Foundation
GNU	A UNIX-like computer operating system that is free software and contains no UNIX code
GPL	GNU General Public License
IP	Intellectual property
LGPL	GNU Lesser General Public License
MIT	Massachusetts Institute of Technology
ML	Machine learning
MPL	Mozilla Public License
NDA	Non-disclosure agreement
NGO	Non-governmental organization
OD	Open data

OH	Open hardware
OIN	Open Invention Network
OKFN	Open Knowledge Foundation
OSD	Open Source Definition
OSH	Open source hardware
OSHWA	Open Source Hardware Association
OSI	Open Source Initiative
OSS	Open source software
SaaS	Software as a service
TDM	Text-and-data mining
UNIX	A family of multitasking, multiuser computer operating systems
ZOOM	'3Os and IP awareness raising for collaborative ecosystems' (ZOOM) Project, Grant Agreement No. 101007385

List of Figures

Figure 1. Categories of Free and Nonfree Software	33
Figure 2. Common FOSS Licenses Compatibility Chart	42

List of Tables

Table 1. Principles of Open Source License	17
Table 2. Principles of Open Source License	21
Table 3. Requirements Related to the Distribution of Open Work	25
Table 4. Open License's Required Permission	25
Table 5. Open License's Acceptable Conditions	26
Table 6. Open Source Definition	32
Table 7. Classification of Free Software Licences	37
Table 8. Common Types of CC Licenses	38
Table 9. Common Types of CC Licenses	41
Table 10. Patent Grants Terms and Defensive Termination Provisions of Recommended Free Software Licenses	45
Table 11. Comparison of Common Free Software Licenses	59
Table 12. GNU Affero General Public License 3.0	62
Table 13. GNU General Public License 3.0	63
Table 14. GNU Lesser General Public License 3.0	64
Table 15. Mozilla Public License 2.0	65
Table 16. Apache Software License 2.0	66
Table 17. MIT	66
Table 18. Example of Issues and Appropriate Solutions as Part of a Compliance Program	68
Table 19. List of Licenses Conformant with the Definition of 'Open'	104

1.Executive Summary

1.1. Project Outline

ZOOM aims to raise awareness on the importance of intellectual property (IP) generation and management in collaborative innovation ecosystems which rely on three key assets: software, hardware, and data. Free and open source software (FOSS), open (source) hardware (OH/OSH) and open data (OD) are essential for a sustainable, trustworthy and sovereign industrial ecosystem. However, lack of competences in matching business models with appropriate licensing frameworks prevents unlocking the full potential of emerging technologies, such as artificial intelligence (AI), blockchain and quantum computing.

Strategic autonomy and technological sovereignty are possible only if we have a sound understanding of how specific business models interact with different licensing models. This hinges upon awareness of the advantages of keeping digital ecosystems open and providing knowledge generators with an informed choice about the appropriate licensing strategy for their business model.

The project aims to reach three core target groups of actors in innovation ecosystems: knowledge generators, innovation support organisations and other key stakeholders (e.g., policy makers):

- Knowledge generators will benefit from tools for learning and supporting everyday development. These tools will increase their competences on open licensing strategies for IP management. They will be helpful in identifying viable business models enabled by open licensing strategies.
- Supporting organisations will get tools for internal training and support material to empower their clients. These tools will identify formats or services that can increase the reach of the IP management.
- Other key stakeholders will benefit from a white book suggesting an integrated EU approach to open licensing of software, hardware, and data as intertwined components of the innovation policy. This document will set the basis for further research on the role of open licensing from a legal and business perspective.

1.2. Purpose of the Document

This document is the first deliverable in Work Package 1 'IP legal framework for open source software, hardware and data' (WP1). It provides an overview of the basics of open source software, open hardware and open data from the perspective of intellectual property law. The document is intended as a background document to the licensing framework that should be developed as a second deliverable in WP1. This deliverable contains summaries of important legal cases and a discussion of important recent cases. Finally, it sets out the legal cases that should provide a focus for the licensing framework, i.e., the problems it should tackle.

2. Open Source as a Development and Licensing Model

2.1. Definition of “Openness” and “Open Source”

The concept of openness describes the idea of sharing and facilitating reuse in domains where access has been restricted by legal or other mechanisms. This is especially the case where scarcity or closure has been introduced in the form of intellectual property¹. Openness is an antidote to such closure as it aims to ‘remove restrictions to use (including modification and reuse) and access’².

The Open Knowledge Foundation has adopted an extensive definition of ‘open’ which captures the spirit of the ‘open everything’ movement.³ In its latest version 2.1, the Open Definition ‘makes precise the meaning of “open” with respect to knowledge, promoting robust commons in which anyone may participate, and interoperability is maximized’⁴. The key points of the definition have been summarised, as follows⁵:

‘Knowledge is open if anyone is free to access, use, modify, and share it — subject, at most, to measures that preserve provenance and openness.’

Openness is therefore defined in terms of freedom of use, study, modification, and sharing⁶. Restrictions to these freedoms are only acceptable if they ‘ensure that material which is released under an open license remains open and attribution is preserved’⁷. As Andrew Katz has noted, there are competing rationales of use-

¹ Andrew Katz, ‘Open Hardware’ in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) 513 <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>.

² *ibid.*

³ The Open Knowledge Foundation is a UK not-for-profit organization, Open Knowledge Foundation, ‘Open Knowledge Foundation Mission’ (*Open Knowledge Foundation*) <<https://okfn.org>> accessed 19 February 2023.

⁴ Open Knowledge Foundation, ‘Open Definition v2.1’ (*Open Definition*, November 2015) <<http://opendefinition.org/od/2.1/en/>> accessed 19 February 2023.

⁵ *ibid.*

⁶ Katz, ‘Open Hardware’ (n 1) 514.

⁷ *ibid.*

maximisation (advocated by the Open Source Initiative) or anti-closure (advocated by the Free Software Foundation) in the conversation on open technology⁸.

In fact, this is what gave birth to the two terms 'open source' and 'free software'. Thus, copyleft emerged as a form of restriction on sharing and reuse. Perhaps surprisingly, copyleft itself can be seen as a form of restriction because business customers may be reluctant to use technology licensed under a copyleft license for fear of non-compliance and the (wrong) assumption that source code which incorporates copyleft code cannot be commercially exploited, and the concern of losing control on the code once opened. In contrast, permissive licenses (e.g., Apache 2.0) emerged to promote use-maximization, which thus requires that 'the material in question can be used by as many people as possible and for as many purposes as possible, without discrimination'⁹.

Open source was coined by a group of pragmatic technologists who feared the rhetoric of the free software movement could discourage adoption of open source software by business. The term 'open source' was an attempt to reconcile the two differences between the two approaches.

There is not one uniform definition of 'open source', but the concept can be seen in two main ways: as a development methodology and as a licensing model. The term originated in the field of software development precisely because of the value that this development model gives to software¹⁰.

The essence of this model is given in Eric Raymond's article 'The Cathedral and the Bazaar'¹¹. In his article, Raymond relates the development of proprietary software to the building of a cathedral and open source development to a bazaar. Like proprietary software, the building of a cathedral requires a strong organization (e.g., the Church) to conceive a project, raise funds and find a good builder and a team of artisans. The project's progress depends on the organization's available funds and the builder's skill to supervise the project. Unlike proprietary software, open source software development looks more like a bazaar where anybody is free to sell their merchandise. It is the market forces that decide what is bought and sold. Importantly, in a free market development is collaborative with virtually unlimited resources and no single controlling organization. As a result, the direction of a project may change at any given

⁸ ibid 515.

⁹ ibid.

¹⁰ Heather Meeker, *Open (Source) for Business: A Practical Guide to Open Source Software Licensing - Third Edition* (Third Edition, Independently published 2020) 6.

¹¹ Eric Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (O'Reilly 1999).

moment, if the marketplace wishes so, and any project can grow into many other projects¹².

As a development model, open source is flexible in that it can adapt to the needs of the market. This vision has been criticized by free software advocates as incompatible with the philosophy of the free software movement. However, practically every market resorts to standardization sooner rather than later. This means that the 'forking' of either software projects or licensing terms is poised to cease naturally once certain projects or licensing terms become standard in the community¹³. As Meeker reveals, '[o]pen source ... is a world in which freedom is paramount but some practices are discouraged – not by fiat but by consensus'¹⁴.

It is open source as a development model that gives the inherent value of open source software. Essentially, open source software is expressed in source code and it is this subject matter that links open source as a development model to open source licensing and compliance¹⁵. Achieving the goals of open source as a development model depends on an array of legal devices that allow developers control the distribution of source code. These are intellectual property rights that subsist in the different subject matter that may be developed and released as open source, such as computer programs or computer-aided design (CAD) files. Intellectual property rights, specifically copyright, give human-readable source code the protection which delineates mere information from executable knowledge¹⁶. The origins of open source as a licensing model therefore lies in intellectual property rights.

Different subject matter is protected by different intellectual property rights. For example, the expressive elements of computer programs are protected by copyright but if they have technical functional aspects, these may be protected by patents. Hardware designs may be protected by registered or unregistered design rights, but potentially also by other rights. Similarly, data may be covered by database rights or other domain-specific rights. We therefore need to distinguish between the subject matter and the intellectual property rights that subsist in it. Naturally, this also has

¹² Meeker, *Open (Source) for Business* (n 10) 6.

¹³ *ibid* 6–7.

¹⁴ *ibid* 7.

¹⁵ Mirko Böhm, 'Economics of Open Source' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) 301–302 <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>.

¹⁶ *ibid*.

implications for the open source licensing model that applies to the subject matter at hand.

While we have well-crafted and tested licences for open source software, this is not necessarily the case for hardware or data. That's why we have different definitions of open source for software, hardware, and data. While they share much of the spirit of open source as a development model, they are profoundly different when it comes to the licensing of the underlying intellectual property rights.

2.2. Open Source Software

In spite of the competing philosophies behind free software and open source software, the definitions of the two reveal similarities. The free software definition begins with the four essential software freedoms¹⁷, namely:

- The freedom to run the program as you wish, for any purpose (freedom 0)
- The freedom to study how the program works and change it so it does your computing as you wish (freedom 1)
- The freedom to redistribute copies so you can help others (freedom 2)
- The freedom to distribute copies of your modified version to others. By doing this, you can give the whole community a chance to benefit from your changes. (freedom 3)

The Open Source Definition (OSD), stewarded by the Open Source Initiative (OSI) takes a more pragmatic, as opposed to philosophical, approach. The OSD makes it clear that '[o]pen source doesn't just mean access to the source code'¹⁸. The definition formulates 10 principles, which must all be met for a license to be called 'open source'. Crucially, any license that does not conform to even a single principle of the OSD should not be considered an 'open source' license.

These principles are summarized in the following table:

¹⁷ Free Software Foundation, 'What Is Free Software?' (*GNU Project*) <<https://www.gnu.org/philosophy/free-sw.html>> accessed 19 February 2023.

¹⁸ Open Source Initiative, 'The Open Source Definition' (*Open Source Initiative*, 7 July 2006) <<https://opensource.org/osd/>> accessed 19 February 2023.

Table 1. Principles of Open Source License

Principle	Content
1. Free Redistribution	The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
2. Source Code	The program must include source code and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a pre-processor or translator are not allowed.
3. Derived Works	The license must allow modifications and derived works and must allow them to be distributed under the same terms as the license of the original software.
4. Integrity of The Author's Source Code	The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
5. No Discrimination Against Persons or Groups	The license must not discriminate against any person or group of persons.
6. No Discrimination Against Fields of Endeavor	The license must not restrict anyone from making use of the program in a specific field of endeavour. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
7. Distribution of License	The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. License Must Not Be Specific to a Product	The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
9. License Must Not Restrict Other Software	The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
10. License Must Be Technology-Neutral	No provision of the license may be predicated on any individual technology or style of interface.

As discussed above, the need of openness arises where there is closure in the form of IP. Computer programs are protected by copyright law as literary works. Furthermore, functional aspects of software that meet the criteria for patentability may also be protected by patents. Keeping control over software in source or binary form therefore depends on intellectual property rights, specifically copyright and patents¹⁹. Any open source license, therefore, relies on the private law mechanisms of licenses and contracts²⁰. These mechanisms allow rightsholders to authorise certain acts, such as use or distribution, that are otherwise prohibited by the default IP rules.

If a rightsholder (e.g., developer) chooses not to pick a license for the distribution of their software, this means that the software is not released as open source software. As the GitHub policy on licensing a repository clarifies, 'without a license, the default copyright laws apply, meaning that you retain all rights to your source code and no one may reproduce, distribute, or create derivative works from your work'²¹. Similarly, allowing computer programs to enter the public domain is not an alternative to the challenges of applying copyright law to open source. This is so because the category of public domain depends on surrendering of legal rights, some of which may be inalienable in some jurisdictions (e.g., moral rights)²².

Legally, open source software is defined by the categories of ownership and licensing²³. This means that any rightsholder who wants to enable use of source code, such as modification or (re)distribution, must rely on licenses or contracts. Software can be dual licensed under both proprietary (i.e., non-open source) and open source license²⁴. However, if a developer wants their software to be considered open source, then it must be released under an open source software license that meets the OSD.

¹⁹ Ian Walden, 'Open Source as Philosophy, Methodology, and Commerce: Using Law with Attitude' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) 2

<<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>.

²⁰ *ibid* 3.

²¹ GitHub, 'Licensing a Repository' (*GitHub Docs*)

<<https://ghdocs-prod.azurewebsites.net/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/licensing-a-repository>> accessed 19 February 2023.

²² Walden (n 19) 15–18.

²³ Amanda Brock, 'Business and Revenue Models and Commercial Agreements' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) 330 <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>.

²⁴ *ibid* 331.

While software can be released under both an open source and a proprietary license, licenses are either open source or not.

Finally, unlike the other two types of subject matter, open source software has been the subject of several court cases interpreting various aspects of open source licensing and the nature of open source licenses. The most relevant of these cases are summarized in the appendix to this deliverable.

2.3. Open Hardware

Unlike open source software, open source hardware (OSH) is a less mature and clear concept whose boundaries are still being defined. There are at least three distinct topics that are relevant in the context of open hardware²⁵:

- Open hardware movement has taken a more cautious approach compared to open source software
- There are fewer specific licenses for open hardware and existing open source software licenses are not always a good match
- Application of IP rights to different aspects of hardware is still unclear.

The first hurdle is to define what is covered by ‘hardware’. While copyright law has accommodated computer programs under the heading of ‘literary works’, there is no equivalent category for hardware. This is partly owing to the lack of a single definition of hardware²⁶.

Originally, the term ‘open hardware’ was used in the late 1990s to refer to hardware with freely available interface information. Access to the hardware designs or modification rights was therefore not required. At the time, the point was to facilitate device driver programming and ensure that more devices could interoperate with the GNU/Linux ecosystem²⁷.

Hardware is fundamentally different to software because it consists of physical material, i.e., atoms²⁸. Under this broad definition hardware can take many different forms, such as printed circuit boards, silicon chip designs, mechanical devices, and even artistic objects. Since any hardware design includes electronic components (e.g.,

²⁵ Katz, ‘Open Hardware’ (n 1) 490.

²⁶ *ibid* 491.

²⁷ *ibid*.

²⁸ *ibid* 490–491.

capacitors, transistors, resistors) described at a high level of abstraction, the OSH design cannot be replicated as easily as an OSS design²⁹.

Significant drivers of adoption were the maker movement, various meetups and groups, and of course the work of the Open Source Hardware Association (OSHW), which is further discussed in the primer on open hardware licensing. We now have a host of examples of successful open hardware projects, such as Arduino, RepRap, open source chip designs based on RISC-V, the CERN White Rabbit Project, OpenSPARC, SiFive, etc.

The most widely popular definition of 'open hardware' is the Open Source Hardware Definition and the associated Statement of Principles adopted by OSHWA. The Statement of Principles defines open source hardware as 'hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximise the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs'³⁰.

Like the general definition of 'open', the 'open source hardware' definition is also based on the Open Source Definition for Open Source Software. Similarly, the definition consists of an introductory section, followed by 12 criteria that distribution terms must meet.

The introductory section clarifies that 'Open Source Hardware (OSH) is a term for tangible artifacts — machines, devices, or other physical things — whose design has been released to the public in such a way that anyone can make, modify, distribute, and use those things'³¹. Importantly, it also highlights the difference between hardware and software, as follows: 'Hardware is different from software in that physical resources must always be committed for the creation of physical goods. Accordingly,

²⁹ Blind, Knut and others, 'The Impact of Open Source Software and Hardware on Technological Independence, Competitiveness and Innovation in the EU Economy' (Publications Office of the European Union 2021) Final Study Report 338 <<https://ec.europa.eu/newsroom/dae/redirection/document/79021>>.

³⁰ Open Source Hardware Association, 'Open Source Hardware (OSHW) Definition 1.0' (*Open Source Hardware Association*, 26 May 2012) <<https://www.oshwa.org/definition/>> accessed 20 February 2023.

³¹ *ibid.*

persons or companies producing items (“products”) under an OSHW license have an obligation to make it clear that such products are not manufactured, sold, warranted, or otherwise sanctioned by the original designer and also not to make use of any trademarks owned by the original designer’³².

The distribution terms for Open Source Hardware must comply with the following criteria:

Table 2. Principles of Open Source License

Criterion	Content
Documentation	The hardware must be released with documentation including design files and must allow modification and distribution of the design files. Where documentation is not furnished with the physical product, there must be a well-publicized means of obtaining this documentation for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The documentation must include design files in the preferred format for making changes, for example the native file format of a CAD program. Deliberately obfuscated design files are not allowed. Intermediate forms analogous to compiled computer code — such as printer-ready copper artwork from a CAD program — are not allowed as substitutes. The license may require that the design files are provided in fully documented, open format(s).
Scope	The documentation for the hardware must clearly specify what portion of the design, if not all, is being released under the license.
Necessary software	If the licensed design requires software, embedded or otherwise, to operate properly and fulfil its essential functions, then the license may require that one of the following conditions are met: <ul style="list-style-type: none"> a) The interfaces are sufficiently documented such that it could reasonably be considered straightforward to write open source software that allows the device to operate properly and fulfil its essential functions. For example, this may include the use of detailed signal timing diagrams or pseudocode to clearly illustrate the interface in operation. b) The necessary software is released under an OSI-approved open source license.
Derived Works	The license shall allow modifications and derived works and shall allow them to be distributed under the same terms as the license of the original work. The license shall allow for the manufacture, sale, distribution, and use of products created from the design files, the design files themselves, and derivatives thereof.
Free redistribution	The license shall not restrict any party from selling or giving away the project documentation. The license shall not require a royalty or other fee for such sale. The license shall not require any royalty or fee related to the sale of derived works.

³² *ibid.*

Attribution	The license may require derived documents, and copyright notices associated with devices, to provide attribution to the licensors when distributing design files, manufactured products, and/or derivatives thereof. The license may require that this information be accessible to the end-user using the device normally but shall not specify a specific format of display. The license may require derived works to carry a different name or version number from the original design.
No Discrimination Against Persons or Groups	The license must not discriminate against any person or group of persons.
No Discrimination Against Fields of Endeavor	The license must not restrict anyone from making use of the work (including manufactured hardware) in a specific field of endeavour. For example, it must not restrict the hardware from being used in a business, or from being used in nuclear research.
Distribution of License	The rights granted by the license must apply to all to whom the work is redistributed without the need for execution of an additional license by those parties.
License Must Not Be Specific to a Product	The rights granted by the license must not depend on the licensed work being part of a particular product. If a portion is extracted from a work and used or distributed within the terms of the license, all parties to whom that work is redistributed should have the same rights as those that are granted for the original work.
License Must Not Restrict Other Hardware or Software	The license must not place restrictions on other items that are aggregated with the licensed work but not derivative of it. For example, the license must not insist that all other hardware sold with the licensed item be open source, nor that only open source software be used external to the device.
License Must Be Technology-Neutral	No provision of the license may be predicated on any individual technology, specific part or component, material, or style of interface or use thereof.

Open hardware distribution may trigger an array of IP rights which are not typical for distribution of software. These may be patent rights that impinge on the use and exploitation of a piece of hardware. Unlike software, where it is challenging to prove infringement, a piece of hardware may infringe a patent in a much more obvious way. Hardware may also concern registered and unregistered design rights, database rights, and semiconductor topography (mask) rights. Furthermore, copyright may subsist in the documentation and other related content that meets the criterion of being an author's own intellectual creation.

2.4. Open Data

The roots of opening data for reuse lies in public sector data³³, open government data³⁴ and open science³⁵.

In Europe, the regulatory basis for the reuse of public sector information stems from 2003. The current directive, Open Data Directive, on open data and the re-use of public sector information is the Directive 2019/1024 of the European Parliament and of the Council of 20 June 2019³⁶. It bases on the PSI Directive, i.e., Directive 2003/98/EC of the European Parliament and of the Council³⁷, amended by the Directive 2013/37/EU of the European Parliament and of the Council³⁸, that sets a constant basis and increasing demands for re-use of public sector data.

When looking into the licensing of open data, the key aspect is to understand the differences between licensing of open source software and open hardware compared to licensing of open data. All of these have a clear difference in the basis of their IP rights. Data as such is not protected by copyright, additional categories of IP rights apply for instance to databases. In addition, technological advancements affect the ways data is used, and there is a clear trend towards building products and services upon data. All these aspects affect the way open data is or should be licensed. The differences in the IP rights basis and trends affecting the licensing of open data are studied in more detail below in Section 6.1.

A good starting point for the definition of open data, is the Open Knowledge Foundation³⁹ (OKFN) Open Definition. OKFN Open Definition can be summarized as:

³³ 'From the Public Sector Information (PSI) Directive to the Open Data Directive | Shaping Europe's Digital Future' (15 February 2023) <<https://digital-strategy.ec.europa.eu/en/policies/psi-open-data>> accessed 14 March 2023.

³⁴ 'Open Government Data - OECD' <<https://www.oecd.org/gov/digital-government/open-government-data.htm>> accessed 14 March 2023.

³⁵ 'Open Science | UNESCO' <<https://www.unesco.org/en/open-science>> accessed 14 March 2023.

³⁶ 'EUR-Lex - 32019L1024 - EN - EUR-Lex' <<https://eur-lex.europa.eu/eli/dir/2019/1024/oj>> accessed 14 March 2023.

³⁷ 'EUR-Lex - 32003L0098 - EN - EUR-Lex' <<https://eur-lex.europa.eu/eli/dir/2003/98/oj>> accessed 14 March 2023.

³⁸ 'EUR-Lex - 32013L0037 - EN - EUR-Lex' <<https://eur-lex.europa.eu/eli/dir/2013/37/oj>> accessed 14 March 2023.

³⁹ OKFN is a not-for-profit organization with a mission of: "an open world, where all non-personal information is open, free for everyone to use, build on and share; and creators and innovators are fairly recognized and rewarded." 'Mission' <<https://okfn.org>> accessed 14 March 2023.

“Open means anyone can freely access, use, modify, and share for any purpose (subject, at most, to requirements that preserve provenance and openness).”⁴⁰

The goals of the OKNF Open Definition include promoting a robust commons in which anyone may participate⁴¹, compatibility and interoperability is maximized, and quality of data ensured⁴². The OKNF Open definition defines openness with regard to two categories; firstly data and secondly content⁴³. When open data is useful, in the form of being usable and becomes used, it builds up and becomes open knowledge⁴⁴. Even though the OKNF Open definition covers both categories, it is important to distinguish the two, as their IP rights basis is different.

The OKFN Open Definition is aligned with the OSI OSS Definition regarding software and the OSHWA Definition regarding hardware. This essential meaning matches that of “open” with respect to software as in the Open Source Definition and is synonymous with “free” or “libre” as in the Free Software Definition and Definition of Free Cultural Works.⁴⁵

OKFN Open Definition requires from an *open work* the following requirements relating to its distribution⁴⁶:

⁴⁰ ‘The Open Definition - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge’ <<http://opendefinition.org/>> accessed 14 March 2023.

⁴¹ ‘Open Definition 2.1 - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge’ <<http://opendefinition.org/od/2.1/en/>> accessed 14 March 2023.

⁴² ‘The Open Definition - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge’ <<http://opendefinition.org/>> accessed 14 March 2023.

⁴³ *ibid.*

⁴⁴ ‘What Is Open?’ <<https://okfn.org>> accessed 14 March 2023.

⁴⁵ ‘Open Definition 2.1 - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge’ <<http://opendefinition.org/od/2.1/en/>> accessed 14 March 2023.

⁴⁶ *ibid.*

Table 3. Requirements Related to the Distribution of Open Work

Requirements	Content
Open license or status	The work must be in the public domain or provided under an open license (as defined in Section 2). Any additional terms accompanying the work (such as a terms of use, or patents held by the licensor) must not contradict the work's public domain status or terms of the license.
Access	The work must be provided as a whole and at no more than a reasonable one-time reproduction cost, and should be downloadable via the Internet without charge. Any additional information necessary for license compliance (such as names of contributors required for compliance with attribution requirements) must also accompany the work.
Machine readability	The work must be provided in a form readily processable by a computer and where the individual elements of the work can be easily accessed and modified.
Open format	The work must be provided in an open format. An open format is one which places no restrictions, monetary or otherwise, upon its use and can be fully processed with at least one free/libre/open-source software tool.

In order to qualify as *open license* in accordance with OKFN Open Definition, the *license* needs to satisfy the following conditions⁴⁷:

Table 4. Open License's Required Permission

Required permissions	Content
Use	The license must allow free use of the licensed work.
Redistribution	The license must allow redistribution of the licensed work, including sale, whether on its own or as part of a collection made from works from different sources.
Modification	The license must allow the creation of derivatives of the licensed work and allow the distribution of such derivatives under the same terms of the original licensed work.
Separation	The license must allow any part of the work to be freely used, distributed, or modified separately from any other part of the work or from any collection of works in which it was originally distributed. All parties who receive any distribution of any part of a work within the terms of the original license should have the same rights as those that are granted in conjunction with the original work.
Compilation	The license must allow the licensed work to be distributed along with other distinct works without placing restrictions on these other works.

⁴⁷ *ibid.*

Non-discrimination	The license must not discriminate against any person or group.
Propagation	The rights attached to the work must apply to all to whom it is redistributed without the need to agree to any additional legal terms.
Application for any purpose	The license must allow use, redistribution, modification, and compilation for any purpose. The license must not restrict anyone from making use of the work in a specific field of endeavour.
No charge	The license must not impose any fee arrangement, royalty, or other compensation or monetary remuneration as part of its conditions.

Table 5. Open License's Acceptable Conditions

Acceptable conditions	Content
Attribution	The license may require distributions of the work to include attribution of contributors, rights holders, sponsors, and creators as long as any such prescriptions are not onerous.
Integrity	The license may require that modified versions of a licensed work carry a different name or version number from the original work or otherwise indicate what changes have been made.
Share-alike	The license may require distributions of the work to remain under the same license or a similar license.
Notice	The license may require retention of copyright notices and identification of the license.
Source	The license may require that anyone distributing the work provide recipients with access to the preferred form for making modifications.
Technical restriction prohibition	The license may require that distributions of the work remain free of any technical measures that would restrict the exercise of otherwise allowed rights.
Non-aggression	The license may require modifiers to grant the public additional permissions (for example, patent licenses) as required for exercise of the rights allowed by the license. The license may also condition permissions on not aggressing against licensees with respect to exercising any allowed right (again, for example, patent litigation).

3. Principles of Free and Open Source Software Licensing

3.1. Introduction

Any attempt to compile information on principles of Free Software (also known as Open Source Software) licensing bares the responsibility of dealing with an extraordinarily rich social, legal policy and economic experience achieved at global scales in the last forty years. FOSS is considered by some by the most impactful driver of innovation in the world today⁴⁸. The supportive social movement which has been pushing forward the ideals of free exchange of knowledge, creative expression, collaborative development and respect to digital commons, has grown to become a foundational element not only to economy at digital age⁴⁹, but mainly to the notion of democracy in the information society⁵⁰.

Free Software can be found in every spot of the digital world. Together with open standards⁵¹, they are present in explicit domains like apps and programs used in laptops, mobile phones or IoT devices, but also less obviously in servers, digital

⁴⁸ C. Herstatt and D. Ehls (2015). *Open Source Innovation: The Phenomenon, Participant's Behaviour, Business Implications*. Routledge.

⁴⁹ Yochai Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom* (Yale University Press 2006)
<https://www.benkler.org/Benkler_Wealth_Of_Networks.pdf>.

⁵⁰ See, for example, Matthias Kirschner, 'Democracy Requires Free Software' (*Free Software Foundation Europe*, 2010) <<https://fsfe.org/freesoftware/democracy.en.html>>; Vassilis Kostakis, 'The Advent of Open Source Democracy and Wikipolitics: Challenges, Threats, and Opportunities for Democratic Discourse.' (2011) 7 *Human Technology*; Michael Dizon, 'Free and Open Source Software Communities, Democracy and ICT Law and Policy' (2010) 18 *International Journal of Law and Information Technology*.

⁵¹ Open Standards' definition has varied along the years, but general classification also differentiates between "open" and "proprietary". For concepts relating Open Standards to FOSS, see: FSFE. Open Standards - Overview. Website, available at: <https://fsfe.org/freesoftware/standards/index.en.html>; Phipps, Simon (2022). Open Source and Open Standards: The Unseen Conflict. Talk at SFSCon, Bolzano, available at: <https://www.sfscon.it/talks/open-source-and-open-standards-the-unseen-conflict/>. For an in-depth analysis on the history of Open Standards, see: A Russell, *Open Standards and the Digital Age: History, Ideology, and Networks* (Cambridge University Press 2014); Laura Di Nardis (ed), *Opening Standards the Global Politics of Interoperability* (MIT 2011).

libraries and APIs, operating system kernels, encryption, and security applications. The intense deployment of Free Software has led invariably to questions and issues that has permeated public discussions in the last decades spanning over diverse fields such as economics, philosophy, sociology, law, political and business sciences.

This chapter focuses, however, on the legal premises in which the Free Software movement rides upon. The success of the Free Software movement was facilitated by the formation, development, and consolidation of a series of legal and policy innovations concerning software licensing. Free Software licenses were key to boost the large adoption and collective development of software. These processes translated into the migration of the core development and sharing environments to the outskirts of the restrictive corporate silos, where innovation follows pre-determined commercial interests⁵². The cross-sector and cross-industry engagement among business actors, individuals and the public sector for use, re-use, implementation and re-implementation of Free Software required competent legal instruments that disciplined a large spectrum of transactions that enabled the practical transit of the necessary applied knowledge, setting the terms for rights waive and retention accordingly to established principles of sharing elected by the communities around the globe.

Free Software licenses are the cornerstone where rights and obligations regarding software usage and distribution, access to source code and its correlated sharing and improvement are settled. Therefore, the main objective of this chapter lays on the simple and accessible presentation of the foundations of Free Software licensing. For that, the characterisation of the two principal types of Free Software licenses non-reciprocal and copyleft, as well as on license compatibility will be central. Besides, due to its impact and importance to Free Software licensing, software patents and export control regulations will also be addressed in an introductory manner.

Since Free Software coincidentally with open data enjoys a rich history of analyses, studies and research, this chapter dedicate efforts in providing in-depth information on relevant and contemporary works related to the topics tackled by this deliverable.

⁵² For the diverse definitions of innovation and the background driving factors, see Ariel Ezrachi and Maurice Stucke, *How Big-Tech Barons Smash Innovation and How to Strike Back* (Harper Collins Publishers 2022).

3.2. Copyright and Frameworks for Reuse of Software

In the EU legislation, software is considered an intellectual work, and is protected primarily⁵³ by copyright law⁵⁴. Copyright is a legal construct that grants someone exclusive rights over a creative work. The most important exclusive right is in the name: the right to produce copies. Although the EU legislation does not provide a clear definition of what comprises ‘software’ or ‘computer programs’, it sets apart what should be excluded from copyright protection⁵⁵. Ideas and principles which underlie any element of a computer program, including those which underlie its interfaces do not enjoy copyright protection⁵⁶. Copyright makes software non-shareable by default⁵⁷. Diverse legal mechanisms and instruments, such as licenses, agreements and contracts operate within established frameworks of copyright, patent, contract, and trade secrets law to transfer rights and set obligations accordingly desired outcomes. In this sense, a license is an authorization to use, release, or distribute someone else’s rights over a determined work which can be a piece of text, a song,

⁵³ Copyright is the primary source because software can be subject of other areas of public and private law, such as patent law, trade secrets and contract law.

⁵⁴ The WIPO Copyright Treaty (1996) is a special agreement under the Berne Convention on Copyright which deals with the protection of works and the rights of their authors in the digital environment. In addition to the rights recognized by the Berne Convention, they are granted certain economic rights. The Treaty also deals with two subject matters to be protected by copyright: (i) computer programs, whatever the mode or form of their expression; and (ii) compilations of data or other material (“databases”). See WIPO Copyright Treaty, art 4. The first EU Directive on the legal protection of computer programs was Council Directive 91/250/EEC of 14 May 1991. The most recent version is Directive 2009/24/EC.

⁵⁵ The European Court of Justice (ECJ) examined the scope of the term in *SAS Institute Inc v World Programming Ltd* [2012] ECLI:EU:C:2012:259 (Court of Justice of the European Union) [14], holding that a ‘computer program’ does not extend to the functionality of a program, the programming language, or the format of data files.

⁵⁶ For an in-depth analysis on the how the copyright “expression vs idea” dichotomy applies to software, see Noam Shemtov, *Beyond the Code: Protection of Non-Textual Features of Software* (Oxford University Press 2017)

<<https://oxford.universitypressscholarship.com/10.1093/oso/9780198716792.001.0001/oso-9780198716792>> accessed 13 May 2022.

⁵⁷ Unless in case of copyright statutory exceptions. Articles 5 and 6 of Software Directive (2009/24/EC) provide some of such exceptions related to back-up, private study and decompilation. However, copyright exceptions lack full harmonization among EU members. National implementations of the various exceptions and limitations to copyright and related rights vary in extend and scope. For a broader analysis of how exceptions proposed by other Directives have been introduced particularly by EU member states, check the Copyright Exceptions initiative., accessible at <https://copyrightexceptions.eu/>.

or software⁵⁸. Licenses and license agreements are legal documents that define how content can be used, modified, and shared. Software licenses inform how the rights holder wants the software to be used under which freedoms or restrictions. The absence of a license does not necessarily make a software freely available to use. Unless it has been explicitly and validly placed in the public domain, or falling under copyright exception, using code without a license may be considered copyright infringement.

The Free Software movement kicked-off in late 1980s⁵⁹ the fundamental distinction between “proprietary” and “Free Software”. While both terms relate to license terms, proprietary software became what could be considered the opposite of Free Software. Proprietary software is referred to the one distributed under restrictions that prevent users from enjoying the “four freedoms” (more on these below), concepts considered the philosophical pinnacle of the movement. Such distinction grew in importance as the curation of the term “Free Software” institutionalized with the emergence of the Free Software Foundation in 1985. The distinction became crucial to differentiate, for example, software that would be distributed gratis but could be considered proprietary. At the same time, the term “commercial software” has been carelessly used to refer to proprietary software. However, Free Software does not exclude commercial use.

3.3. Defining Free and Open Source Software

As previously mentioned, both “Free Software” and “Open Source Software” are curated terms respectively by the Free Software Foundation (the FSF)⁶⁰ and Open

⁵⁸ Historically, there has been a long standing debate on whether the FOSS licenses are regarded only as *licenses per se* or contracts. Such distinction may have consequences on remedies for license infringement, as copyright law presume injunction, contract law can be limited to damages and profits. In the EU a German case, *Welte / Sitecom Deutschland GmbH* [2004] Az 21 O 6123/04 (LG München). it was held that failing to comply with the terms of a GPL license constituted both breach of contract as well as copyright. For further considerations, see Miriam Ballhausen, ‘Copyright Enforcement’ in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en>>. For different doctrines and interpretations, see GA Rub, ‘Copyright Survives: Rethinking The Copyright-Contract Conflict’ (2017) 103 Virginia Law Review 1141.

⁵⁹ For historical insights, see Steven Weber, *The Success of Open Source* (Harvard Univ Press 2005).

⁶⁰ The Free Software Foundation is a North American non-profit corporation supporting free software development. Founded in 1985 by Richard Stallman, the FSF sponsors the GNU Project—the

Source Initiative (the OSI)⁶¹. In contrast with software under a proprietary license – where only a few rights are granted, such as the right to use a program but not to distribute copies of it – the term “Free Software” refers to software that enables users to enjoy four essential freedoms: to use, to study, to share, and to improve the software.

- **Freedom to use:** Installing and running the program should not be restricted. Placing restrictions on the use of Free Software, such as time ("30 days trial period", "license expires January 1st, 2024"), purpose ("permission granted for research and non-commercial use", "may not be used for benchmarking"), or geographic area ("must not be used in country X") makes a program proprietary.
- **Freedom to study.** Access to source code should not be restricted. Placing legal or practical restrictions on the comprehension or modification of a program makes it proprietary. For example, imposing purchase of special licenses, signing of a Non-Disclosure-Agreement (NDA) or - for programming languages that have multiple forms or representation.
- **Freedom to redistribute copies.** Sharing can be done for monetary charge. Commercialization is possible. Free software opposes “non-commercial.” Free Software must be available for commercial use, development, and distribution.
- **Freedom to share improved copies.** Any user obtaining a copy of the software, who has complied thus far with the conditions of the free license covering the software can modify, improve, and publicly share the software.

The term “Open Source Software” was originally based on the Debian Free Software Guidelines⁶². Currently is curated by the OSI, which states that Open Source doesn't just mean access to the source code, but also that the distribution terms must comply with the following criteria:

ongoing effort to provide a complete operating system licensed as free software. The FSF publishes and curates the GPL family of licenses. More at: <https://www.fsf.org/about/>.

⁶¹ The OSI is a North American public benefit corporation founded in 1998 by Bruce Perens and Eric S. Raymond. The organization maintains the OSI License Review Process to “ensure that licenses and software labelled as open source conform to existing community norms and expectations”. More at: <https://opensource.org/approval>.

⁶² The Debian Free Software Guidelines is a set of guidelines developed by the Debian Project to determine whether a piece of software can be included in the operating system “Debian”. The first version of the guidelines was published in 1997. More at: https://www.debian.org/social_contract#guidelines

Table 6. Open Source Definition

Open Source Definition
<ol style="list-style-type: none"> 1. Free Redistribution: The license shall not require a royalty or other fee for the distribution by the licensee. 2. Source Code must be included 3. Derived Works must be permitted 4. Integrity of The Author's Source Code: The license may require derived works to carry a different name or version number from the original software. 5. No Discrimination Against Persons or Groups 6. No Discrimination Against Fields of Endeavour 7. Distribution of License: The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties. 8. License Must Not Be Specific to a Product 9. License Must Not Restrict Other Software 10. License Must Be Technology-Neutral

The interrelation between the two terms of “Free Software” and “Open Source” has a historically charged experience in which departing positions from ethical or operational points of view polarized the debate⁶³. In legal terms, however, they are usually understood as synonyms, as both refer to specific development and licensing models which characteristics should abide the definition above⁶⁴. Nevertheless, Free and

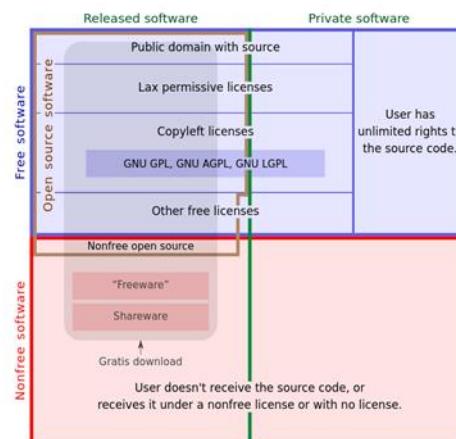
⁶³ On the terminological aspect, see Björn Schießle, ‘Free Software, Open Source, FOSS, FLOSS - Same but Different - FSFE’ (*FSFE - Free Software Foundation Europe*, 8 December 2012) <<https://fsfe.org/freesoftware/comparison.html>> accessed 26 February 2023.

⁶⁴ For instance, Meeker, *Open (Source) for Business* (n 10).

Open Source Software are not synonymous with “public domain”⁶⁵, “freeware, shareware or freemium”⁶⁶ and “source available software”⁶⁷.

Figure 1. Categories of Free and Nonfree Software

Categories of Free and Nonfree Software



The diagram in Figure 1 explains the different categories of software⁶⁸.

⁶⁵ While software in the public domain certainly can overlap with the aims of Free Software, as a rule, Free Software is not synonymous with public domain. The public domain consists of all the creative work (including software) to which no copyright applies. The rights to these works may have expired, been expressly waived, or may be inapplicable. While these are the very general principles behind the public domain, the decisive factor of what constitutes public domain will always be determined by the legal principles in the country in which a work is to be used. This is because different national jurisdictions often differ in how they apply fundamental elements of copyright, which can have the effect of impacting what constitutes the public domain in that country. Creative Commons has created a license to explicitly dedicate work as close as possible into the public domain, the Creative Commons CC0 1.0 Universal (CC0) License. See for more information Juan Carlos De Martin and Rosnay Mélanie, *The Digital Public Domain Foundations for an Open Culture* (Open Book Publishers 2012).

⁶⁶ “Freeware” and “freemium” are concepts related to economic price of software, differently from FOSS that contains an ethical and political dimension. Freeware refers to software, most often proprietary, that is distributed at no monetary cost to the end user. Every publisher defines its own rules for the freeware it offers. See “Freeware Definition”. *The Linux Information Project*. 2006. Shareware is software which comes with permission for people to redistribute copies but says that anyone who continues to use a copy is *required* to pay a license fee. Freemium is a pricing strategy by which a basic product or service is provided free of charge, but money (a premium) is charged for additional features, support or goods that expand the functionality of the program.

⁶⁷ Source-available software is software released through a source code distribution model that includes arrangements where the source can be viewed, and in some cases modified, but without necessarily meeting the criteria to be called open-source.

⁶⁸ FSF. *Categories of Free and Nonfree Software*. Diagram licensed under Creative Commons CC-BY-4.0 license. No modifications were made.

3.4. Types of Free Software Licenses

In the late 1980s, two alternative ways of achieving the objectives of the Free Software philosophy were developed: the reciprocal (copyleft) and non-reciprocal (permissive) licensing. Both models acquired numerous variants, but the main differences have been referring to how software can be relicensed.

Free Software licensing stewardship is not a centralized activity. Individuals, business, communities and public agencies have written FOSS licenses according to their contextual needs. However, license proliferation concerns have been an agenda within the Free Software community. Indeed, the diverse types of Free Software licenses can negatively affect the whole ecosystem through the introduction of increasingly complex license selection, license interaction, and license compatibility considerations⁶⁹. Both FSF⁷⁰ and OSI⁷¹ maintain extensive lists of software licenses based on the Free and Open Source definitions to determine whether licenses meet its standards in order to qualify as a “Free Software” or “Open Source” license. The OSI, differently from the FSF, provides a public and formal process for license review, validation and approval⁷².

3.5. Reciprocity of Free Software Licenses

Historically, the Free Software community has been using terms as “permissive” and “copyleft” to classify licenses according to the rights granted or retained by the copyright owner. Although the concept of “license reciprocity” has been understood as synonym of “copyleft”, it refers to “control on the works’ derivatives and to ensure

⁶⁹ See more at OSI. *The License Proliferation Project*. OSI. Available at: <https://opensource.org/proliferation>

⁷⁰ For the FSF, free software licenses are subdivided in two categories: ‘GPL-compatible’ and those that are ‘GPL-incompatible’. GPL compatibility is an important criterion, as the FSF promotes the GPL as the optimal license in general.

⁷¹ The OSI keeps a list of approved licenses and licenses which have been superseded or retired. More at: <https://opensource.org/licenses-draft>

⁷² The OSI’s review process is openly accessible to anyone. Applicants should inform the rationale for the submitted license, compare to and contrast with the most similar OSI-approved license(s), describe any legal review the license has been through, and provide results of any legal analysis if available. Besides, submission must recommend which license proliferation category is appropriate. For more details, see: <https://opensource.org/approval>

those improvements”⁷³. Therefore, some have argued that, instead choosing grants of rights to classify Free Software licenses, taking reciprocity as main parameter would allow a better understanding of the community expectations over the license terms. In this way, the main distinction among types of Free Software licenses would concern the expected scope of the reciprocity by the parties submitted to the license terms. Communities have norms based of reciprocal behaviour and expect determined sharing behaviour with their code based on the same freedoms they have received in the first place⁷⁴. In this sense, Free Software licenses can be categorized as:

- Non-reciprocal (permissive)
- File-scoped reciprocal licenses (weak copyleft)
- Project-scoped reciprocal (strong copyleft).

3.5.1. Non-reciprocal Licenses (Permissive)

Non-reciprocal licenses (also commonly known as permissive or non-copyleft) impose on downstream users of the code lax obligations, mainly concerned to license notice. The scope of obligations can largely vary depending on the objectives of the license, but the range of obligations are generally limited. Differently from copyleft licenses, non-reciprocal licenses allow the code to be used in proprietary derivative works. In such cases, the derivative work would still be proprietary, notwithstanding that it contains code from a source that is covered by a FOSS license. Examples include the Apache 2, the MIT, and BSD licenses.

3.5.2. File-scoped Reciprocal Licenses (Weak Copyleft)

Conversely, file-scope reciprocal licenses, like the Mozilla Public License, set the scope of the reciprocity to the individual files within the project, not the whole project altogether. File change or reuse must be licensed the same way as the original file, but there are no requirements placed on other files combined in new projects. Similar

⁷³ Patrice-Emmanuel Schmitz, ‘Licence Compatibility, Permissivity, Reciprocity and Interoperability’ (*European Commission - JoinUp - EUPL*, 2023) <<https://joinup.ec.europa.eu/collection/eupl/licence-compatibility-permissivity-reciprocity-and-interoperability>>.

⁷⁴ Simon Phipps, ‘Permissive and Copyleft Are Not Antonyms’ (*Open Source Initiative Website*, 2017) <<https://opensource.org/node/875>>.

to project-scoped reciprocal licenses, the distribution of binaries requires the availability of source code of those respective binaries, and additional restrictions on the exercise of the license are not allowed.

Regarding the LGPL, the differences among file-scoped and project-scoped reciprocities are clear. The GNU Project has two principal licenses to use for libraries. One is the GNU Lesser GPL (LGPL); the other is the ordinary GNU GPL. The LGPL permits use of the library in proprietary programs, differently from the GPL. The reciprocity among them is different. The LGPL, which traditionally is considered a weak copyleft license, allows combination of the resulting binary with non-GPL-licensed works (unlike the GPL itself). This is different from the Mozilla Public license, which is also considered a “weak copyleft license”. LGPL is not “weak” in the same way MPL is, for example. Code from an LGPL project itself is fully reciprocally licensed at a project level. Any code borrowed from it for other uses as well as any alternative uses of the project itself are expected to be fully licensed under the same LGPL. Within the project itself, LGPL is “strong copyleft” just like GPL code, but the resulting executable does not necessarily have “strong copyleft” requirements – it’s effectively non-reciprocal in many uses⁷⁵.

3.5.3. Project-scoped Reciprocal Licenses (Strong Copyleft)

Licenses like GPL family of licenses and EUPL set the scope of the expected reciprocity to include any code needed to create the resulting project. The derivative and/or combined work has to be licensed under the same license as the original work. The strongest form of reciprocity can place conditions on the licensing of all the other code compiled together to make the eventual binary executable program and require that all modifications of a program (i.e., any derivative work) be licensed under the original license. This is the main difference from file-scoped reciprocity. Nevertheless, similarly to file-scoped reciprocal licenses, the distribution of binaries requires the availability of source code of those respective binaries, and additional restrictions on the exercise of the license are not allowed.

⁷⁵ *ibid.*

Table 7⁷⁶ below summarizes the three-layered classification delineated above.

Table 7. Classification of Free Software Licences

Non-reciprocal licenses (Permissive)	File-scoped reciprocal licenses (Weak copyleft)	Project-scoped reciprocal licenses (Strong copyleft)
Examples: MIT, Apache 2.0	Examples: MPL, EUPL	Examples: GPL, LGPL, AGPL
<ul style="list-style-type: none"> – No restriction to code's use – (Sometimes) copyright notice requirement 	<ul style="list-style-type: none"> – No restriction to code's use – Copyright notice requirement – Along the binaries, source code must be available – Source code must be available under the same copyleft terms – No restrictions on the exercise of the license – Combined code may be not submitted to copyleft 	<ul style="list-style-type: none"> – No restriction to code's use – Copyright notice requirement – Along the binaries, source code must be available – Source code must be available under the same copyleft terms – No restrictions on the exercise of the license – Combined code may be submitted to copyleft

3.6. Other Types of Licenses in Software Projects

Although Free Software licenses can be used for non-software works, there are licenses that suit better other purposes, like documentation, database, or hardware. Creative Common licenses are one example of dedicated license for data and content. Initiatives regarding AI and hardware produce licenses that incorporate more adequate terms than software licenses.

3.6.1. Licenses for Documentation, Data, Image, and Video

The re-use of copyrighted works, be it data, image, or video, should be authorized by license. As it occurs with software licensing, however, license proliferation can pose a threat for practical reuse of works. Creative Commons licenses and public domain tools provides a free and standardized way to grant permissions for copyrighted works, to ensure proper attribution, and allow copy, distribution, and reuse of those

⁷⁶ Summary based on Heather Meeker, *Open (Source) for Business: A Practical Guide to Open Source Software Licensing* (Third Edition, Independently published 2020) 33–34.

works⁷⁷. Creative Commons licenses are not designed for software, but rather for other kinds of creative works such as websites, scholarship, music, film, photography and literature. Such content can also be incorporated in software projects as video games, for example. Although there are six types of CC licenses, the three below are more encountered in Free Software projects.

Table 8. Common Types of CC Licenses

	CC BY	CC BY-SA	CC0
Main features	Requires attribution of reused work	Requires attribution and same license for modified material (similar to copyleft)	Public domain dedication tool
Grants	Sharing, adapting, reusing	Sharing, adapting, reusing	Sharing, adapting, reusing
Obligations	Appropriate credit to author No additional restriction	Appropriate credit to author Share-Alike: redistribution should be at same licensing terms No additional restriction	No patent or trademark rights granted

3.6.2. Licenses for Data to Be Used in AI-powered Software

The “AI revolution” is based on the enormous capacity of current data processing used for specific machine and deep learning techniques. Data may be consumed, transformed, and incorporated into AI models in ways that are different from how software and other creative content are used. Commonly used Free Software or Creative Commons licenses may encounter difficulties by being used as input to train AI-powered computer models. As *Creative Commons* explains⁷⁸: “*There is no consensus on whether the use of copyright works as input to train an AI system is an exercise of an exclusive right. There remains significant legal uncertainty about whether copyright applies to AI training, which means it may not always be clear*

⁷⁷ For an in-depth analysis of CC licenses and work, see Simoni Aliprandi and Carlo Piana, *A Complete Manual with a Theoretical Introduction and Practical Suggestions* (2012). See also the Creative Commons Licenses website: <https://creativecommons.org/about/cclicenses/>

⁷⁸ See for more information Brigitte Vezina and Sarah Pearson, ‘Should CC-Licensed Content Be Used to Train AI? It Depends.’ (*Creative Commons*, 2021) <<https://creativecommons.org/2021/03/04/should-cc-licensed-content-be-used-to-train-ai-it-depends/>>.

whether a CC license applies. In other words, there is no consensus on whether the use of copyright works as inputs to train an AI system is an exercise of an exclusive right (e.g., reproduction, adaptation, etc.). The situation is likely to vary across jurisdictions, as countries progressively regulate the copyright-AI nexus. In the US, the use of works to train AI is likely considered fair use. In the EU, Article 3 of the Directive on Copyright in the Digital Single Market (DSM) provides an exception for non-commercial text-and-data mining (TDM, a form of AI) by research and cultural heritage institutions, while Article 4 offers an exception regime for commercial TDM, from which rightsholders may opt-out". Traditional software licenses for software and creative content might not apply in expected ways to open data.

The Linux Foundation attempted in 2017 to address license agreement that would clearly enable use, modification, and open data sharing, with a particular eye to AI applications with the Community Data License Agreement (the CDLA)⁷⁹. In its first version, the CDLA provided grant of rights for recipients of data to use, share and modify the data for any purpose. It also permitted using the results from analysed data to create AI and ML models, without the obligations to sharing the data itself. It was launched with two initial types: a non-reciprocal variant (permissive), with attribution-style obligations, and a reciprocal (share-alike, copyleft) variant, with reciprocal commitment regarding resharing the raw data. The second version of the CDLA, launched in 2021, simplifies the license text and keeps only the non-reciprocal variant. The CDLA Version 2 could be compared to non-reciprocal Free Software licenses, such as the MIT or BSD-2-Clause licenses, albeit specific to data and with even more limited obligations⁸⁰.

3.6.3. Licenses for Hardware

Software, information, and content can remain in the abstract domain as intangible assets. The concept of hardware, however, can cover everything from printed circuit boards to silicon chip designs, to cases for computer hardware, to mechanical

⁷⁹ See more at Linux Foundation (2017). *Community Data License Agreement*. Website, available at: <https://cdla.dev/>

⁸⁰ Linux Foundation, 'Enabling Easier Collaboration on Open Data for AI and ML with CDLA-Permissive-2.0' (2021) <<https://www.linuxfoundation.org/press/press-release/enabling-easier-collaboration-on-open-data-for-ai-and-ml-with-cdla-permissive-2-0>>.

devices⁸¹. The Open Source Hardware (OSH) definition⁸² is a term for machines, devices, or other physical artefacts whose design has been released to the public in such a way that anyone can make, modify, distribute, and use those devices. Dedicated open hardware licenses, such as the CERN Open Hardware License family⁸³ and the TAPR Open Hardware License⁸⁴ seek to apply diverse forms of reciprocity to hardware licensing in similar fashion as software ones.

3.7. Free Software License Compatibility

When a software project combines two pieces of code, or merges code from one into another, the issue of whether the licenses of each software or code would allow or prohibit this combination. If the licenses allow it, the licenses can be said to be “compatible” with one another. Compatible licenses ensure that the code under one Free Software license can be combined with code under another, and the resulting software can be distributed under either Free Software license without violating the terms of the other. License compatibility is a legal framework that allows for pieces of software with different software licenses to be distributed together. Incompatible licenses may result in copyright infringement.

Proprietary licenses are generally program-specific and incompatible; authors must negotiate to combine code. While non-reciprocal licenses (permissive) are generally compatible with each other, as they impose very lax or no obligations on downstream users, reciprocal licenses (copyleft) licenses provide that derivative works may only be used under the same license conditions, copyleft licenses are only compatible with other Free Software licenses when the other license does not contain any license requirements that are not provided by the compatible copyleft license, or the other

⁸¹ See more in Katz, ‘Open Hardware’ (n 1).

⁸² The open source hardware statement of principles and definition were developed by members of the Open Source Hardware Association. See more at: <https://www.oshwa.org/definition/>. Open Source Hardware Association (n 30).

⁸³ The CERN Open Hardware Licence version 2 consists of three variants: A strongly-reciprocal variant, CERN-OHL-S, a weakly-reciprocal variant, CERN-OHL-W and a permissive variant, CERN-OHL-P. More at: CERN (2020). *CERN OHL version 2 An Introduction and Explanation*. Available at: https://ohwr.org/project/cernohl/wikis/uploads/0be6f561d2b4a686c5765c74be32daf9/CERN_OHL_rationale.pdf

⁸⁴ The TAPR Open Hardware License is “The Tomorrow’s Ham Radio Technology Today” contribution to the community of Open Hardware developers. More at: <https://tapr.org/the-tapr-open-hardware-license/>

license contains a special compatibility or opening clause. The table⁸⁵ below illustrates the differences in obligations of reciprocal and non-reciprocal licenses. These differences affect directly license compatibility.

Table 9. Common Types of CC Licenses

Non-reciprocal licenses	Scenario	Obligations
	If you distribute the code	You must provide license notice
Reciprocal licenses	Scenario	Obligations
	If you distribute in binary form	You must make the corresponding source code available
	If you distribute in source code form	The obligation is fulfilled
	Inbound = Outbound	You must relicense the software you received on the same copyleft terms

License compatibility is a complex issue and in larger projects require due diligence procedures for conformity⁸⁶. Standardization of license terms unarguably makes the diligence process easier and avoids license proliferation. The reciprocity expected in FOSS licensing determines the general principle that inbound (from contributors) and outbound (to other contributors and users) licensing terms and obligations should match. The FSF classifies licenses on the basis whether they are or not compatible with the GPL⁸⁷. The chart below illustrates the terms and conditions of the four classes of licenses⁸⁸:

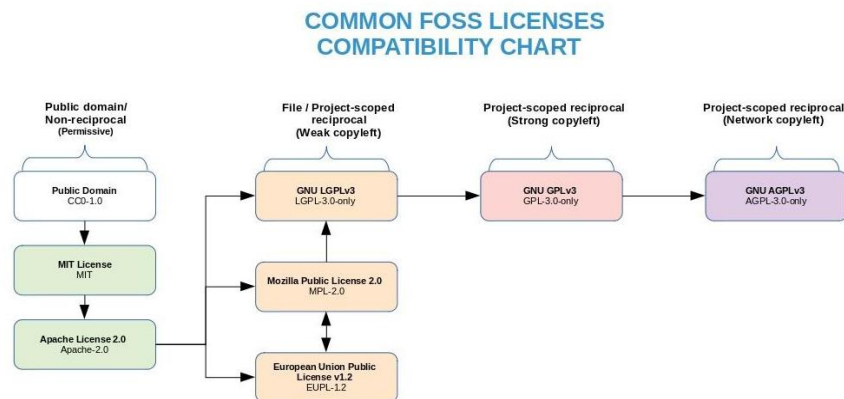
⁸⁵ Heather Meeker, 'Open Source Software Licensing Basics for Corporate Users' (2020) <<https://www.youtube.com/watch?v=gF4b1TA5Q5w>>.

⁸⁶ For more on the concepts of horizontal and vertical compatibility, see Chapter 4 – License Compatibility in Meeker, *Open (Source) for Business* (n 76) 63–75.

⁸⁷ See FSF. *Various Licenses and Comments about Them*. FSF website, available at: <https://www.gnu.org/licenses/license-list.html>

⁸⁸ Chart based on David Wheeler, 'The Free-Libre / Open Source Software (FLOSS) License Slide' <<https://dwheeler.com/essays/floss-license-slide.html>>. The information was licensed under Creative Commons "Attribution-Share Alike 3.0 License". Modifications of the content: The classification of the licenses was altered, new licenses were added, such as EUPL and AGPL. The chart has been colorized.

Figure 2. Common FOSS Licenses Compatibility Chart



How to use this chart: To see if a software can be combined, start at their respective licenses and follow to the license boxes that can be reached by the arrows. An arrow from box A to box B means that you can combine software with these licenses; the combined result effectively has the license of B, possibly with additions from A.

The Joinup platform of the European Commission offers a Joinup Licensing Assistant (JLA) tool⁸⁹ whose functionality enables one to determine the compatibility between any inbound license (covering third party source code that one plans to use in their project) and an outbound license (already covering the main project source code and/or planned for distribution of the project). This tool makes finding, comparing and selecting software licenses based on their content seamless. The JLA tool checks the compatibility based on many crucial classifications like obligations, permissions, prohibitions, interoperability, laws and support.

3.8. Free Software Licensing and Software Patents

A patent is a type of state granted monopoly on rights for an invention that enables the patent holder to exclude other people from practicing the invention or idea claimed in the patent. While copyright protects the expression of ideas, patents work to protect the ideas themselves, not just the expressions of them. The main effect of a patent is therefore to give patent holders the right to challenge any use of the invention or the

⁸⁹ See more at JLA - Compatibility Checker. Available at: <https://joinup.ec.europa.eu/collection/eupl/solution/joinup-licensing-assistant/jla-compatibility-checker>

idea by another person. A software patent is a patent on an invention which is implemented in a piece of software or in a combination of software and hardware. In contrast to copyright (which is exercised over the code itself), software patents are exercised over the functionality in which the software is intended to work. In principle it is independent of the code.

Historically the Free Software community has taken vehement positions against software patents⁹⁰. Patent law varies fundamentally among countries. European patents are governed by the European Patent Convention, a treaty between all member states of the European Union and several other European countries⁹¹. Co-existing with the European patent system, national patent laws in Europe offer additional opportunities for patent application at the national level. A new patent system called the Unitary Patent System is in place to start from June 2023 in the European Union⁹². This system aims to implement unified patent recognition in up to 25 EU Member States, as well as a Unified Patent Court that will handle all disputes related to unitary patents. There are concerns that having such a centralized and unitary system would further stifle the spirit of sharing and cooperation inherent in Free Software, if software patents are allowed to exist in this system⁹³.

The Free Software community has been raising the negatives issues software patents cause to the whole Free Software environment. Under copyright law applicable to software, if part of the software code infringes on someone else's copyright, this issue can be in principle resolved by rewriting the offending section of code, while continuing to ensure that the underlying function that the code aims to achieve remains the same. However, in the case of a software patent, it does not matter how the code is written, or what programming language is used, as the patent acts as a blanket restriction against anyone but the patent holder from implementing a certain idea. Accordingly,

⁹⁰ See Philip Leith, *Software and Patents in Europe* (1st edn, Cambridge University Press 2007) <<https://www.cambridge.org/core/product/identifier/9780511495267/type/book>> accessed 26 February 2023; *Patent Absurdity* (Directed by Luca Lucarini, Free Software Foundation 2010) <<https://patentabsurdity.com/about.html>>; Sebastian Haunss, *Conflicts in the Knowledge Society: The Contentious Politics of Intellectual Property* (Cambridge University Press 2013); Meeker, *Open (Source) for Business* (n 76) 63–75.

⁹¹ There is no consensus among legal scholars and policy makers about the definition of a software patent in Europe. According to Art. 52 (1) of the European Patent Convention, “European patents shall be granted for any inventions which are susceptible of industrial application, which are new, and which are not obvious.” Therefore, software patents add legal risks, and as a result thereof, costs to software development.

⁹² EPO. *Unitary Patent*. Available at: <https://www.epo.org/applying/european/unitary/unitary-patent.html>

⁹³ Panos Alevropoulos, ‘The Threat of Software Patents Persists’ (*Free Software Foundation*, 2021) <<https://endsoftwarepatents.org/2021/08/the-threat-of-software-patents-persists/>>.

once a patent holder accuses a software project of infringing a patent, the project must either stop implementing the offending feature, or expose the project and its users to legal risk. The GPLv3 states in its preamble that every program is threatened constantly by software patents. This is due to the patent system allowing enforcement of a patent not requiring the patent owner to engage in any business he/she is trying to protect. This is why patent “trolls” can exist – companies that do not engage in any business except suing other for patent infringement⁹⁴.

Free Software licenses are primarily copyright licenses, but many of them have expressed and implied patent licensing built into them. While traditional patent licenses usually seek to slice the patent rights granted into narrow pieces, Free Software licenses are purposely broad, mainly because if someone distributes Free Software and holds software patents, an absurd situation may arise where parties may use, study improve and share the software, but simultaneously be prohibited from using the software by the software patent. FOSS licenses in general contain two kinds of patent provisions: grant of patent rights and defensive termination provisions.

3.8.1. Granting Patent Rights

A Free Software license may require that distributors of a software give recipients a license to use any necessary patents. Some licenses contain expressed grant but other grant patent rights by implication. Even in the absence of an express patent grant, all Free Software licenses grant patent rights in some extension. A Free Software patent license grant has only one field limitation, which is that the right is granted only in connection with the exercise of the copyright granted for the software. Any other field limitation, such as territory, commercial or technology fields is against Free Software and Open Source definition, and therefore not included.

3.8.2. Retaliating in Case of Patent Aggression

A Free Software license may have the effect of making patent aggression less attractive by revoking patent rights that any aggressor received through the license. A

⁹⁴ Companies spend more money for their legal department, to register patents, to negotiate patent cross-licensing, and to defend themselves against patent claims, making the entire process complicated and expensive. See EFF, ‘Software Patents and Rise of Patent Trolls’ (*Electronic Frontier Foundation*, 2013) <<https://www.eff.org/de/deeplinks/2013/02/deep-dive-software-patents-and-rise-patent-trolls>>.

defensive termination revokes the patent grants in case if someone exercising the license brings a claim accusing the licensor of patent infringement. There are also initiatives to minimize patent aggression towards the Free Software community, for instance the Open Invention Network (OIN)⁹⁵, which enforces a license agreement among organizations in support of patent non-aggression and for free access to OIN's patents.

The table⁹⁶ below summarizes the patent grants terms and defensive termination provisions of the recommended Free Software licenses.

Table 10. Patent Grants Terms and Defensive Termination Provisions of Recommended Free Software Licenses

License	Grant Patent Rights	Termination Trigger	Rights Terminated
GNU GPLv3	Section 11 Express grant <i>Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.</i>	Section 10 <i>You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may (...) initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.</i>	Section 8 <i>All rights can be terminated, including any patent licenses granted.</i>
European Union Public License v1.2	Clause 2 Express grant <i>The Licensor grants to the Licensee royalty-free, non-exclusive usage rights to any patents held by the Licensor, to the extent necessary to make use of the rights granted on the Work under this License.</i>	Clause 5 <i>The Licensee (becoming Licensor) cannot offer or impose any additional terms or conditions on the Work or Derivative Work that alter or restrict the terms of the License.</i>	Clause 12 <i>The License and the rights granted will terminate automatically upon any breach by the Licensee of the terms of the License.</i>
GNU AGPLv3	Section 11 Express grant <i>Each contributor grants you a non-exclusive, worldwide,</i>	Section 10 <i>You may not impose any further restrictions on the</i>	Section 8 <i>All rights can be terminated, including any patent licenses</i>

⁹⁵ Established in 2005, Open Invention Network (OIN) is the world's largest patent non-aggression community and free defensive patent pool. See more at: <https://openinventionnetwork.com/>

⁹⁶ Table built according to Meeker, *Open (Source) for Business* (n 76) 167–169.

	<i>royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.</i>	<i>exercise of the rights granted or affirmed under this License. For example, you may (...) initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it</i>	<i>granted.</i>
GNU LGPLv3 <i>Note: This license incorporates the terms and provisions of GNU GPLv3.</i>	Section 11 (GNU GPLv3) Express grant <i>Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.</i>	Section 10 (GNU GPLv3) <i>You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may (...) initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.</i>	Section 8 (GNU GPLv3) <i>All rights can be terminated, including any patent licenses granted.</i>
Mozilla Public License 2.0	Clause 2.1 Express grant <i>Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license: under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.</i>	Clause 5.2 <i>If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent.</i>	Clause 5.2 <i>The rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.</i>
Apache License 2.0	Clause 3 Express grant <i>Each Contributor grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free,</i>	Clause 3 <i>If You institute patent litigation against any entity (including a cross-claim or counterclaim in a law suit) alleging that the</i>	Clause 3 <i>Any patent licenses granted to You under this License for that.</i>
MIT License	Implied grant <i>The MIT license does not include an express patent license. However, there are opinions that supports the interpretation of implied</i>	General termination conditions <i>No copyright notice and permission notice included in all copies or substantial portions of the Software</i>	

	patent grant provision ⁹⁷ : <i>The permission is here granted to deal in the software without restriction, and to permit persons to whom the Software is furnished to do so (...)</i>		
--	--	--	--

3.9. Free Software Licensing and Export Control Rules

The software development process in environments involving Free Software is, in most of the times, a cross-boundaries activity. A project can receive contributions from developers around the globe. Making Free Software products available in several jurisdictions require sometimes additional steps to satisfy obligations under local laws. Exporting controlled technology requires an extensive compliance program, and depending on the technology involved can be burdensome and complex⁹⁸. This chapter provides only a limited introduction and should not serve as substitute for seeking professional legal advice.

Export control relates to rules that regulates the export of technology considered potentially harmful for the interests contrary of the exporting country. These items are considered to be “controlled”. Because of their technical characteristics or composition, a number of goods can be used not only in the civil domain but also for military purposes. Such goods are termed “dual-use items”. They include, in particular, items from the fields of sensitive electronics, telecommunications and information technology, but also data processing software or other technologies (such as design drawings). Such goods may be the subject of restrictions within the European Union, even when transmitted by online media. Export controls are rules constantly updated and changed to accommodate evolving security risks and rapid developments in

⁹⁷ Scott Peterson, ‘Why so Little Love for the Patent Grant in the MIT License?’ (*opensource.com*, 23 March 2018) <<https://opensource.com/article/18/3/patent-grant-mit-license>> accessed 26 February 2023.

⁹⁸ See, for example Michael Cheng and Mishi Choudhary, ‘Chapter 12 - Export Control’ in Amanda Brock (ed), *Open Source Law, Policy and Practice* (2nd edn, Oxford University Press 2022); Gabriel Ku Wei Bin and others, ‘EU Export Control Regime and Free Software’ <<https://download.fsfe.org/NG10/V2/FSFE%20-%208%20-%20EU%20Export%20Control%20Regime%20and%20Free%20Software.pdf>>.

technology. The export control regime in the European Union has been revised and consolidated in September 2021 by the Regulation (EU) 2021/821⁹⁹.

3.9.1. Dual-use Items and Export Control

Dual-use items are goods, software and technology that can be used for both civilian and military applications. Items listed in Annex I¹⁰⁰ of the Regulation (EU) 2021/821 are considered dual-use goods¹⁰¹. Dual-use items may be traded freely within the EU, except for some particularly sensitive items, which transfer within the EU remains subject to prior authorization (see Annex IV of the Regulation). The list of dual-use goods is updated regularly¹⁰². If the Free Software project develops such technologies and plans to go online, it is necessary to contact the competent national authority of the respective country. Exporting to third countries outside the EU, the section referring to export authorizations contain information on requirements.

3.9.2. EU Export Control Regime

Applicable in all EU countries, the EU export control regime is governed by Regulation (EU) 2021/821 which set up a regime for the control of exports, transfers, brokering, technical assistance, transit, and transfer of dual-use items. This Regulation incorporates the international export control regimes which are at the origin of the control list implemented in the EU¹⁰³. In certain cases, EU countries may put extra controls on non-listed dual-use items because of public security or human rights

⁹⁹ Regulation (EU) 2021/821 of the European Parliament and of the Council of 20 May 2021 setting up a Union regime for the control of exports, brokering, technical assistance, transit and transfer of dual-use items (recast) 2021 (OJ L 206, 1162021).

¹⁰⁰ The Annex I's text is available here: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2021:206:FULL&from=EN#BK_L_20210519-007EN002_ANNEX

¹⁰¹ An ECCN - Export Control Classification Number - is used to identify items which may be subject to export control. Software categories, including operating, process and encryption, are subject to export controls in the EU are listed in Category 5 Part 2 of Annex I of the EU Regulation (EU) 2021/821. Software is listed in the sub-category "D".

¹⁰² On 6 January 2022, the European Commission published the Delegated Regulation (EU) 2022/1, updating the list of dual-use items contained in Annex I to Regulation (EU) 2021/821.

¹⁰³ As the Australia Group, the Wassenaar Arrangement, the Nuclear Suppliers Group, the Missile Technology Control Regime and The Chemical Weapons Convention. More at the EC's dedicated website for export control:
https://policy.trade.ec.europa.eu/help-exporters-and-importers/exporting-dual-use-items_en

considerations. Under the EU regime, controlled items may not leave the EU customs territory without an export authorization.

3.9.3. Export Authorizations and Software

Exporters intending to export or transfer any software or technology must bear the responsibility to check whether its products are on the export list in order to establish whether the exporter is subject to a permit requirement. Licenses are issued by National Authorities¹⁰⁴. Under the EU export control regime, controlled items may not leave the EU customs territory without an export authorization¹⁰⁵. According to the Regulation (EU) 2021/821 dual-use items shall mean items, including software and technology, which can be used for both civil and military purpose. An extensive definition of terms such as ‘software’ and ‘technology’ can be found at the end of the list of goods in Annex I of the Regulation. Software categories, including operating, process and encryption, are subject to export controls in the EU are listed in Category 5 Part 2 of Annex I. All items on the list require a license for export outside the EU unless they qualify for an exemption.

An export of controlled product (software/technology) occurs when the software is actually shipped, transferred or transmitted (physically or electronically) out of the EU. In addition, releases/disclosures of software source code to a foreign national in the EU or out of it, and releases/disclosures of encryption source code and technology in a foreign country to a foreign national are also governed by EU export control laws. This covers all means of software sharing, including uploading software to online repositories (like GitHub or GitLab), offering cloud services and sending software via email.

3.9.4. Free Software and EU Export Control Regime

The EU export regime provides exceptions to export license requirements for goods and technology delineated under category 0-9 in the Annex I of the EU Regulation (EU) 2021/82. As for software, export control requirements do not apply to products

¹⁰⁴ EC contact website for export control inquiries:
https://policy.trade.ec.europa.eu/contacts_en#_exports

¹⁰⁵ There are four types of export authorizations. See more in EC's knowledge base on export control:
https://circabc.europa.eu/ui/group/654251c7-f897-4098-afc3-6eb39477797e/library/e7dc5aae-bce0-4f45-b1e5-bb15b272b66b?p=1&n=10&sort=modified_DESC

for user's personal use and the items conforming to the conditions mentioned under the Cryptography Note (also discussed in the following sections) in the Regulation. Controls on technology transfers do not apply to information used for "basic scientific research" or that was released is in the "public domain" or "for minimum necessary information for patent applications". These terms originate from the international export control regimes and are broadly defined on purpose in order to apply to the wide range of technology including Category 5 Part 2 of Annex I of the EU Regulation (EU) 2021/821.

For export control objectives, Free Software can be considered software where the source is made publicly available, i.e., under "public domain" as defined by the Regulation¹⁰⁶. It does not relate solely to the software license (i.e., GPL) but the software must be made available to third parties publicly. Therefore, the current understanding is that "software in the public domain", i.e., software that has been made available without restriction (excluding copyright restrictions) is not subject to export control. Downstream redistributors of modified project code, or products derived from it, where the source code is not publicly available would still need to evaluate their own compliance with the export control laws.

According to the Regulation, "public domain" is defined as means technology or software which has been made available without restrictions upon its further dissemination (copyright restrictions do not remove technology or software from being in the public domain). Information is not in the public domain if it is available to only a restricted group of persons. This includes information that is only made accessible after an individual decision has been taken by the information carrier. In this case, not everyone has the possibility to access the information. It should also be considered that information is only in the public domain once it has been published. The fact that copyright restrictions do not remove technology or software from being in the public domain is important considering that Free Software is distributed under copyright.

3.9.5. Encryption Products and EU Export Control Regime

However, "public domain" exemption does not apply to "information security" items, i.e., controlled cryptography products¹⁰⁷. The so-called "Cryptography Note" –

¹⁰⁶ See Ku Wei Bin and others (n 98).

¹⁰⁷ Strictly, the controls apply to those using symmetric algorithms with a key length over 56 bits or asymmetric algorithms with a key length over 512 bits. The controls are not restricted to hardware but include components, software and technology such as design data. Software or technology

Category 5 Part 2 of Annex I of the EU Regulation (EU) 2021/821 establishes control over software encryption¹⁰⁸. Exporters are responsible to determine if the items are subject to control under the Cryptography note. Exporters are advised to request guidance from the Competent authority or to notify it of the results of their assessment for further confirmation, according to national regulations or practices. Exporters not sure whether the above Cryptography Note applies to their software products, should consider applying for an export license.

3.9.6. Cyber-surveillance Items

The Regulation (EU) 2021/821 in art. 2(20) provides for a definition of ‘cyber-surveillance items’ to mean dual-use items specially designed to enable the covert surveillance of natural persons by monitoring, extracting, collecting, or analysing data from information and telecommunication systems. The Regulation mandates the exporter to seek an authorization for all export of cyber-surveillance items not listed in Annex I that can be used for the commission of serious violations of human rights or international humanitarian law or can pose a threat to international security or the essential security interests of the Union and its Member States, upon becoming aware of it or being informed by the competent authority². The exporter is bound by a notification obligation towards the competent authority if upon due diligence, the exporter becomes aware that the cyber-surveillance items are used for the above-mentioned purposes.

3.9.7. Compliance with EU Export Control Regime

Exporting controlled technology requires an extensive compliance program, and depending on the technology involved can be complex. This chapter provides only a limited introduction and should not serve as substitute for seeking professional legal advice. In case you planning to export encryption Free Software Exporters, you are advised to request guidance from the Competent authority or to notify it of the results

that is carried or transmitted outside the EU (e.g., sent by email or by remote access of a server) is also subject to control.

¹⁰⁸ The cryptographic note is available at:

https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2021:206:FULL&from=EN#BK_L_20210519-007EN002_ANNEX

of their assessment for further confirmation. In this process, the following is important to notice:

- **Keep information public and open.** It is fundamental for projects to make information available transparently and publicly to meet the “public domain” criteria. Information related to security issues can be kept public upon availability of fixes. This kind of information should not be kept confidential only to a disclosure list.
- **Make sure the source code is publicly available.** Making source code available is a requirement from Free Software licenses. But for export control policy, in case available encryption software in object code form is being distributed, then source code should be made available as well.
- **Keep detailed registers or records of the exports.** The Regulation (EU) 2021/821 brings an additional responsibility for all exporters of dual-use items to keep detailed registers or records of their exports, in accordance with the national law or practice in force in the Member State concerned. The retention period for these registers and records is now increased from three to five years.
- **Contact the National Authority for Export Control.** In case projects are planning to export encryption, they are advised to request guidance from the Competent authority or to notify it of the results of their assessment for further confirmation. In case of questions regarding whether encryption exception applies to the Free Software products, a specialized lawyer of National Authority for Export Control should be contacted.

3.10. Conclusion

Free and Open Source Software is a vivid reality in digital ecosystems around the globe. The success of the movement revolutionized how copyright law was applied to software regulation. FOSS license are key instruments used to hand control of your project’s software back to the people, allowing them to use, study, share, and improve the code. Choosing a license is a critical point for Free Software projects, as the license terms have directly impact on how code will be reused, setting expectations over reciprocity of rights and obligations among the communities. License compatibility can be a challenging issue, especially in larger projects where various pieces of codes with different licenses should interact to form the final product. Software patents, although a highly controversial topic within the Free Software

community, is a reality faced by many projects. FOSS licenses present forms to grant patent rights and oppose litigation. Lastly, since Free Software development can be a cross-jurisdiction activity, export control rules may apply, raising a complex issue to be dealt by project maintainers.

4. Primer on Free and Open Source Software Licensing Compliance

From its origins in the 1980's, the Free Software movement has experienced an exponential growth from small collaborative projects to large global corporations with 8-digits turnovers¹⁰⁹. In the last decades, organizations (e.g., governments, companies, universities, NGOs, communities, etc.) have realized that Free Software is a key part of their IT strategy, and want to increase internal implementation or foster participation in FOSS development.

The “traditional” proprietary software industry has followed the behaviour of closing access to source code under restricted environments composed by their own employees or licensed from a third party. Copyright over the code has been closely held, and software has been delivered in binary format to clients. With Free Software, however, source code is kept open. The development workflow is transparent and inclusive where even end users of the software actively contribute to upstream projects. Free Software development model has enabled organizations to think differently about how they procure, implement, test, deploy, and maintain software. In other words, the FOSS development model has been characterized by processes and values of increased intra-team collaboration, continuous integration and testing, and greater user involvement¹¹⁰. This approach is complemented by licenses, which set down the legal terms for source code distribution. This chapter explores basic requirements of the most commonly used FOSS licenses, and introduces the legal, operational and logistical challenge to achieve compliance. In particular, the REUSE specifications will be taken as key to simplify workload for informing copyright and license information. In conformity with the ZOOM editorial line, additional material on the addressed topics is kept in footnotes for those interested in in-depth analysis.

¹⁰⁹ For a historical perspective, see Chris DiBona, Sam Ockman and Mark Stone (eds), *Open Sources: Voices from the Open Source Revolution* (O'Reilly 1999); Dave Neary, '6 Pivotal Moments in Open Source History' ([opensource.com](https://opensource.com/article/18/2/pivotal-moments-history-open-source), 2018) <<https://opensource.com/article/18/2/pivotal-moments-history-open-source>>.

¹¹⁰ Eric Raymond's analogy *The Cathedral and the Bazaar* has been historically recalled as a commentary on proprietary vs. open source software. The “cathedral design” represents proprietary model as a tidy and managed process leading to an inexorable and deliberated endpoint. The result may or may not be beautiful, but it will be deliberate. By contrast, the “bazaar design” portrays the FOSS model as a dynamically effusion which evolves organically by broad participation and experimentation, which may be ultimately effective for the customers. See Raymond (n 11).

The broader Free Software adoption by consumers and producers has sparked considerations on the dominant power of large tech corporations, and their behaviour towards the usage of Free Software in digital devices. In determined circumstances, the power of device manufacturers, vendors and internet platforms can be such in digital markets that they perform a “gatekeeper function”, hampering end-user rights to install Free Software on their devices. Therefore, this chapter concludes with the prognosis of two recent developments that impact the FOSS compliance environment as a whole: the right to repair and the economic regulation of large tech corporations in relation to devices from the FOSS perspective.

4.1. Fundamentals of Free Software Licensing Compliance

The introduction of Free Software licensing is the most significant development in software licensing since software licensing began¹¹¹ because it changed completely the way software is developed. With access to source code and transparent development communities, software providers can reduce development costs while remaining active participants in the development process. In addition, end users of the software can also be active in the development process by contributing directly to upstream projects, rather than passive recipients of what the software vendor delivers to them¹¹².

As demonstrated in Chapter 2, the Free Software movement relies on copyright law and contract law, and to a greater or lesser extent patent law, to govern the conduct of users of their code. The choice of a license is one of the first logical steps at the beginning of a Free Software project. The process of choosing a license differs fundamentally from proprietary. While proprietary software requires dedicated license that should be tailored specifically to the project in question, Free Software licenses are public documents usually drafted by third parties (FSF or OSI, for example) and no negotiation are involved. There are many Free Software licenses available for choosing, depending on the objectives desired by the project. Although some have

¹¹¹ However, as Heather Meeker observes, “*software licensing itself has not been with us very long, and in truth, open source licensing has been around for longer than most people realize. In fact, open source licensing was the original model for software licensing, and proprietary licensing is the newcomer*”. See Meeker, *Open (Source) for Business* (n 76).

¹¹² Such elements are relevant factors for building an IT strategy within an organisation. See Ibrahim Haddad and Brian Warner, ‘Establishing an Open Source Software Strategy: Key Considerations and Tactical Recommendations’.

been written to satisfy the legal needs of a corporation or person, the most commonly used licenses provide sets of terms and obligations that allow them to be classified into three groups: non-reciprocal (permissive), file-scoped reciprocal (weak copyleft) and project-scoped copyleft (reciprocal)¹¹³.

As a principle, anyone can create their own Free Software license to dictate the terms and conditions of use of their work. While proprietary licenses are fundamentally incompatible with each other, the Free Software movement has made efforts to avoid license proliferation to increase legal interoperability and simplify license adoption. The existing well-known licenses are standardized, well-documented and have withstood complex legal issues. The compatibility between the existing well-known licenses can also be easily ascertained by developers or any potential user.

The correct appropriation of Free Software licenses requires it to conform to community norms and expectations. This entails a transparent process of public review of the licenses whilst ensuring non-proliferation of unnecessary or duplicative Free Software licenses. The Open Source Initiative (OSI), in this regard, has established a transparent and time-bound review procedure to ¹¹⁴ a formal request for getting a Free Software license approved or/and retired^[10]. The request should entail a clearly described rationale, legal review or analysis, the proliferation category along with a comparison with any similar OSI approved licenses. All the community discussions regarding the submissions taken place on the License Review mailing list of OSI, whereby the submitter can directly engage by replying to any questions or defending any claims. On another hand the Free Software Foundation (FSF) publishes and curates the GPL family of licenses. A license is classified by FSF according to certain key criteria whether it qualifies as a free software license, or if it a copyleft license and its compatibility with the GPL family of licenses.

The following sections will cover how projects can choose a license, apply their terms and establish a baseline compliance program that can attend the needs for respecting inbound and outbound FOSS licenses in their projects.

4.2. Choosing a FOSS License

The choice of a license is fundamental for any Free Software project, as the terms and obligations have a direct impact on how the software development, commercial

¹¹³ See the chapter on “Reciprocity of FOSS Licenses” for an in-depth analysis of each sub-group.

¹¹⁴ See OSI’s License Review Process. Available at: <https://opensource.org/approval>

and distribution processes depend on the rights granted over the source code. In generic terms, adopters (consumers or producers) seek to achieve one or more of these three objectives¹¹⁵:

- **Sharing improvements.** Having made usage of the source code, they want to make sure their contribution will remain free. Such a group prefers reciprocal licenses (copyleft), as such licenses allow others to do almost anything they want with the contributions, except distributing closed source versions.
- **Keeping the licensing obligations to the simplest terms.** Releasing code under a non-reciprocal license (permissive) allows quick and effortless adoption of the software by users. Such licenses are short and simple. They allow others to do almost anything they want with the software, like making and distributing closed source versions.
- **Community engagement**¹¹⁶. In case the objective is to contribute to a software with a community around it, the license choice is usually dependent on the license preferred by the community. The distribution of the software will be fit for further usage on that group.

Although the OSI and FSF have analysed and approved over 100 Free Software licenses, only a percentage of those licenses are in wide use, and many of the others are variations of¹¹⁷. All Free Software licenses are conditional documents. They grant all the rights of copyright, with no field restrictions or limitations. They do impose conditions on exercise of the license, but these are not limited as to the type of use, location of use, number of copies, as it occurs with proprietary schemes. Therefore, compared to proprietary software licenses, Free Software licenses are easier to comply with. Free Software licenses are direct licensing models. They do not grant any right to grant further sublicenses. If a distributor violates a Free Software license, then although the distributor may lose the rights, downstream recipients do not. For the purpose of this chapter, only the most commonly used Free Software licenses are

¹¹⁵ See, for example, Choose an open source license, available at: <https://choosealicense.com/>

¹¹⁶ For an analysis of FOSS communities, see Ross Gardler and Stephen Walli, 'Evolving Perspective on Community and Governance' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (2nd edn, Oxford University Press 2022).

¹¹⁷ See: Synopsis (2022). *Top Open Source Licenses and Legal Risk for Developers*. Available at: <https://www.synopsys.com/blogs/software-security/top-open-source-licenses/>. Check also the OSI's list of approved licenses: <https://opensource.org/licenses-draft>

analysed. The chart below lists the permissions, conditions, and limitations of such licenses¹¹⁸.

4.3. Common Free Software Licenses

¹¹⁸ The information on the table was kindly derived and adapted from the content of the Choose a License website (<https://choosealicense.com/>). The website content is licensed under the [Creative Commons Attribution 3.0 Unported License](#). Modifications of the content: The website content was displayed in a coloured table; column headers were modified. Instead of colours, “yes” was used to indicate the license characteristics. SPDX identifiers were also included.

Table 11. Comparison of Common Free Software Licenses

License Type	License/SPDX code	Commercial Use? ¹¹⁹	Distribution? ¹²⁰	Modification? ¹²¹	Patent Grant? ¹²²	Disclose Source Code? ¹²³	License and Copyright Notice? ¹²⁴	Same license requirement (copyleft)? ¹²⁵	Changes documented? ¹²⁶	Network use is distribution (SaaS)? ¹²⁷	Liability Limitation? ¹²⁸	Trademark Grant? ¹²⁹	Warranty Limitation? ¹³⁰
		Permissions				Conditions				Limitations			
Project-scoped reciprocal (Strong copyleft)	GNU Public License v3 GPL-3.0-only	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes		Yes
	GNU Affero Public License v3 AGPL-3.0-only	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes
File-scoped reciprocal (Weak copyleft)	GNU Lesser Public License v3 LGPL-3.0-only	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes		Yes
	European Union Public License v1.2 EUPL-1.2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes		Yes
	Mozilla Public License 2.0 MPL-2.0	Yes	Yes	Yes	Yes	Yes	Yes	Yes			Yes		Yes
Non-reciprocal (Permissive)	Apache License 2.0 Apache-2.0	Yes	Yes	Yes	Yes		Yes		Yes		Yes		Yes

e)	MIT License MIT	Yes	Yes	Yes	Yes (Implicit)	Yes	Yes	Yes
Public Domain	CC0 1.0 Universal CC0-1.0	Yes	Yes	Yes			Yes	Yes

-
- 119 The software can be used for commercial purposes.
- 120 The software can be distributed to other people and organizations.
- 121 The software can be modified.
- 122 Contributors (copyright holders) grant patent rights. See Chapter 2 for more detailed information on patent grants.
- 123 Source code (not only the binaries) must be disclosed when software is distributed.
- 124 A copy of the license and copyright notice must be included with the software.
- 125 Modifications must be released under the same or, in some cases, similar licenses.
- 126 Changes to the code must be documented.
- 127 Users that interact with the software through the Internet (SaaS) must have access to the source code.
- 128 The license includes a limitation of liability.
- 129 The license does not grant trademark rights.
- 130 The license does not provide warranty.

The Free Software development model presumes that development is distributed among multiple teams, working in different locations, in a fluid structure that is welcoming to new arrivals and resilient to departures of contributing members. To achieve an efficient working environment, Free Software communities have developed processes where code can be submitted and integrated asynchronously, open communication channels, practices for documenting progress, and workflows designed to integrate small increments attending the principle “release early, release often”¹³¹. Understanding license terms is crucial for a fluid workflow where obligations derived from licenses are well understood and legal risks are accounted into the expectations.

Most Free Software license obligations are triggered by a distribution, like the GPL, MPL, EUPL, but some have obligations specifically triggered by access to software via a computer network (AGPL). For a general overview on the most commonly used Free Software licenses, the license requirements are summarized for the four most common use-cases listed below¹³².

1. Distribution of software with **unmodified** copies in binary form
2. Distribution of software with **modified** copies in binary form
3. Distribution of software with **unmodified** copies in source form
4. Distribution of software with **modified** copies in source form.

¹³¹ See Ibrahim Haddad, ‘Understanding the Open Source Development Model’.

¹³² The information of the tables and the methodology of the use-cases was kindly adapted and derived from 2019 Fintech Open Source Foundation, Jilayne Lovejoy, and other contributors. This document is licensed under the terms of the Creative Commons Attribution-Share-alike (CC-BY-SA 4.0) License. It is offered as-is and as-available, without representation or warranty of any kind, whether express, implied, statutory, or other. The original version of this document is available at <https://github.com/finos-osr/OSLC-handbook/>. Modifications on the content: the use-cases were renamed, the six tables (AGPL, GPL, LGPL, MPL, Apache, MIT) were shortened, the order of the information in the table was altered. Information on “linking” was adapted and provided additional source.

4.3.1. Project-scoped Reciprocal Licenses (Strong Copyleft)

4.3.1.1. GNU Affero General Public License 3.0

Table 12. GNU Affero General Public License 3.0

SPDX identifier	AGPL-3.0-only, AGPL-3.0-or-later, depending on the versioning.			
Description	Based on GPL v3, is dedicated to software used over a network (SaaS for instance). This provision requires that the full source code be made available to any network user (not only distribution of the software).			
License obligations	Use-Cases			
	Unmodified Binary	Modified Binary	Unmodified Source	Modified Source
Provide copy of license	Yes	Yes	Yes	Yes
Retain notices on all files	Yes	Yes	Yes	Yes
Notice of modifications		Yes		Yes
Modifications or derivative work must be licensed under same license		Yes		Yes
Provide corresponding source code	Yes	Yes		
Prohibit circumvention of technological measures that prevent users from exercising rights under the license	Yes	Yes	Yes	Yes
No additional restrictions	Yes	Yes	Yes	Yes
Termination provision	<ul style="list-style-type: none"> – License automatically terminates with non-compliance with the terms of the license. – License terminates in case of litigation claiming use of the program under this license violates a patent. 			
Other terms	Installation provision: Providing information necessary to install modified versions on end-user products.			

4.3.1.2. GNU General Public License 3.0

Table 13. GNU General Public License 3.0

SPDX identifier	GPL-3.0-only, GPL-3.0-or-later, depending on the versioning.			
Description	The most popular reciprocal license. Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.			
License obligations	Use-Cases			
	Unmodified Binary	Modified Binary	Unmodified Source	Modified Source
Provide copy of license	Yes	Yes	Yes	Yes
Retain notices on all files	Yes	Yes	Yes	Yes
Notice of modifications		Yes		Yes
Modifications or derivative work must be licensed under same license		Yes		Yes
Provide corresponding source code	Yes	Yes		
Prohibit circumvention of technological measures that prevent users from exercising rights under the license	Yes	Yes	Yes	Yes
No additional restrictions	Yes	Yes	Yes	Yes
Termination provision	<ul style="list-style-type: none"> – License automatically terminates with non-compliance with the terms of the license. – License terminates in case of litigation claiming use of the program under this license violates a patent. 			
Other terms	Installation provision: Provide information necessary to install modified versions on end-user products.			

4.3.2. File-scoped Reciprocal Licenses (Weak Copyleft)

4.3.2.1. GNU Lesser General Public License 3.0

Table 14. GNU Lesser General Public License 3.0

SPDX identifier	LGPL-3.0-only, LGPL-3.0-or-later, depending on the versioning.			
Description	The file-scoped reciprocal license allows developers and companies to use and integrate a software component released under the LGPL into their own (even proprietary) software without being required by the terms of a strong copyleft license to release the source code of their own components. The use of this license is discouraged by the FSF.			
License obligations	Use-Cases			
	Unmodified Binary	Modified Binary	Unmodified Source	Modified Source
Provide copy of license	Yes	Yes	Yes	Yes
Retain notices on all files	Yes	Yes	Yes	Yes
Notice of modifications		Yes		Yes
Modifications or derivative work must be licensed under same license		Yes		Yes
Provide corresponding source code	Yes	Yes		
Prohibit circumvention of technological measures that prevent users from exercising rights under the license	Yes	Yes	Yes	Yes
No additional restrictions	Yes	Yes	Yes	Yes
Termination provision	License automatically terminates with non-compliance with the terms of the license.			
Other terms	– Library license: Allows use of a "suitable shared library mechanism" (including dynamic linking ¹³³) to combine other code under a			

¹³³ Programming involves a trade-off between memory use and speed. "Linking" allows trade-offs in designing a program. In some languages, like C++, there are different ways to link. When a library is called via *dynamic linking*, the computer must find the routine at execution time, execute it, then flush it from memory and return to the program that called it. With *static linking*, by contrast, the routine is always in memory and is part of the same binary file as the routines that call it. See Meeker, *Open (Source) for Business* (n 76).

	<p>different license, under certain conditions.</p> <ul style="list-style-type: none"> – Installation provision: Provide information necessary to install modified versions on end-user products. – Further library restriction: If you modify the library so that it does not function without data or function supplied by your application, the modified library can only be distributed under the terms of GPL-3.0.
--	---

4.3.2.2. Mozilla Public License 2.0

Table 15. Mozilla Public License 2.0

SPDX identifier	MPL-2.0			
Description	As a file-scoped reciprocal license, it is characterized as a middle ground between that seeks to balance the concerns of proprietary and FOSS developers (usually business). File-level reciprocal means that modifications to any file or new files that contain part of original software are governed by the terms of this license. Larger works may be created by combining covered software with code not governed by this license, so long as there is compliance with this license for the covered software. In other words, a larger work using the licensed work may be distributed under different terms and without source code for files added in the larger work.			
License obligations	Use-Cases			
	Unmodified Binary	Modified Binary	Unmodified Source	Modified Source
Provide license			Yes	Yes
Modifications under same license				Yes
Notices			Yes	Yes
Modifications or derivative work must be licensed under same license		Yes		Yes
Provide source code	Yes	Yes		
Termination provision	<ul style="list-style-type: none"> – License terminates upon failure to comply with license unless certain conditions are met – Any patent claims accusing the software by a licensee result in termination of all licenses to the licensee 			
Versioning	Allows use of covered code under the terms of same version or any later version of the license.			
Other terms	Binary distribution's license: Allow distribute binary versions under a different license, so long as there is no alteration or limitation the			

	recipient's right in the source code under this license.
--	--

4.3.3. Non-reciprocal Licenses (Permissive)

4.3.3.1. Apache Software License 2.0

Table 16. Apache Software License 2.0

SPDX identifier	Apache-2.0			
Description	A non-reciprocal license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.			
License obligations	Use-Cases			
	Unmodified Binary	Modified Binary	Unmodified Source	Modified Source
Provide copy of license	Yes	Yes	Yes	Yes
Notice of modifications		Yes		Yes
Copyright and other Notices			Yes	Yes
Provide source code				
Termination provision	Any patent claims accusing the work by a licensee result in termination of all patent licenses to the licensee.			

4.3.3.2. MIT

Table 17. MIT

SPDX identifier	MIT			
Description	A short and simple non-reciprocal license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.			
License obligations	Use-Cases			
	Unmodified Binary	Modified Binary	Unmodified Source	Modified Source
Provide copy of license	Yes	Yes	Yes	Yes
Provide copyright notice	Yes	Yes	Yes	Yes

Once an appropriate FOSS license has been chosen for a project, the next step is to apply it to the software. While applying the license, a number of considerations should be considered. The next section aims to explore how Free Software licenses should be applied and how to organize a basic compliance program that can be implemented depending on the project's needs.

4.4. Introduction to Free Software Licensing Compliance Program

The first step in applying a license to the software require the identification of all copyright holders of the code in question to ascertain the governing license¹³⁴. The public should be effectively informed by providing a copy of the license inside the project's repository and license notices in source files. In relation to this, the REUSE specifications are key (see further in this section). In case a project uses a work of another project, it is important to check the copyright and license information as reflected in that project and incorporate it into one's own project¹³⁵. If the copyright information is missing in the work to be reused, a copyright notice can be added by oneself into one's own project. If there is no license information provided, that does not mean that the work is in public domain. It in fact, inhibits the ability for anyone to reuse the code. In such a case, it would be prudent to contact the author and ask them to license their work. It is also advised to always check the copyright and license information in the root directory or README files of projects, as sometimes this information is only reflected there.

Maintaining compliance with these different types of Free Software licenses seen a daunting task, but compliance is often more of an operational and logistical challenge

¹³⁴ While doing so, one should be mindful of the legislative interpretation of an "author" in different jurisdictions. Often the author and the copyright holder are the same. However, if the author is being paid by their employer to create a work, the employer is often the copyright holder. One should also take into account situations where a contributor may be deceased or go missing. See WIPO, *Understanding Copyright and Related Rights* (2nd edn, WIPO 2016) <https://www.wipo.int/edocs/pubdocs/en/wipo_pub_909_2016.pdf>.

¹³⁵ In general, the notices that are placed in source files do not have to look exactly like each other, and can take many different forms, as long as the standard requirements are met. However, given the scope of ZOOM, the REUSE specifications are recommended, which can simplify your work and foster the integration and reuse of your software through standardizing how notices are displayed.

than a legal challenge. Achieving compliance requires the proper policies and processes, training, tools, and proper working force that enable a project or organization to effectively use FOSS and contribute back to the community. Governance of Free Software projects involves respecting copyrights, trademarks and patents of their respective holders, complying with license obligations, and honouring community rules of interaction and collective work¹³⁶.

A compliance program depends on the size of the Free Software project, as it could be tailored for small teams of contributors working on a small program or to a large corporation implementing Free Software on their production. In any case, it is considered a best practice among the Free Software community to have a compliance policy in place¹³⁷ covering topics such as compliance processes, code release, guidelines for external contribution, and baseline requirements for contracts and trademarks. Depending on the operational scale of the project, it may not need to cover all topics, but issues faced on the reality of the project should have the priority. Most commonly, projects prioritize license compliance, as the legal risks may involve copyright or license infringement, as the table below summarizes with some examples¹³⁸.

Table 18. Example of Issues and Appropriate Solutions as Part of a Compliance Program

Examples of compliance issues	Types of solution
Integrating proprietary software into FOSS.	Perform source code inspections and require prior approval to include proprietary source code into FOSS components.
Integrating or linking FOSS elements into proprietary software.	Perform code scans and require prior approval to use FOSS. Provide training on how dependencies and libraries should work.
Failure to provide source code (when license requires).	Establish a license compliance checklist.
Failure to include a description of the changes.	Add source code marking as checklist item before releasing source code.
Failure to include copyright, attribution or license notices.	Add milestone in compliance process to verify the different types of notices

¹³⁶ See Ibrahim Haddad, *Open Source Compliance in the Enterprise* (2nd edn, The Linux Foundation 2018) <<https://www.linuxfoundation.org/resources/publications/open-source-compliance-in-the-enterprise>>; Gordon Haff, *How Open Source Ate Software: Understand the Open Source Movement and So Much More* (Springer 2021), Chapter 3.

¹³⁷ See Meeker, *Open (Source) for Business* (n 76), Chapter 11 – Open Source policies.

¹³⁸ Table based on information derived from Haddad (n 137), Chapter 1.

The Free Software policy focuses on review of software based on the license that covers it, and this is the way most companies review their potential use of Free Software. Such policy needs not be complex attempting to cover every possible circumstance. The governance evolves over time to be the collection of documentation that enables the involved people to work¹³⁹:

- Procedures for filling a bug or suggesting a new feature¹⁴⁰.
- Instructions for collaborating and proposing merge requests¹⁴¹.
- Codes of conduct for working behaviour and community engagement¹⁴².

At bigger organizations or businesses, a more comprehensive compliance program and its supporting infrastructure with existing internal processes can be implemented. Processes and policies are important; however, they must be light and efficient so that development teams do not regard them as overly burdensome to the development process. In any case, streamlining Free Software governance can rely upon two elements: a simple and clear compliance policy and a lightweight compliance process. For companies, start-ups or smaller organizations starting to implement such processes, can be challenging to balance the compliance efforts with existing internal processes while meeting deadlines regarding products and services. Staffing should be allocated. However, Free Software compliance matters can be handled by someone with some software engineering and legal training. Such compliance efforts may include¹⁴³:

- Setting the terms of a Free Software strategy/policy. Naming a person or a team responsible for its implementation¹⁴⁴.

¹³⁹ Gardler and Walli (n 116).

¹⁴⁰ For example, Framasoft contact section: <https://contact.framasoft.org/en/#framsoft>; Matrix Specification Proposals. Available at: <https://github.com/matrix-org/matrix-spec-proposals>.

¹⁴¹ For example, Offen – contribution instructions. Available at: <https://github.com/offen/offen#feedback-and-contributions-welcome>.

¹⁴² For example, Debian Code of Conduct: https://www.debian.org/code_of_conduct; Debian Diversity statement: <https://www.debian.org/intro/diversity>;

¹⁴³ See Haddad (n 137), Chapter 2.

¹⁴⁴ For a theoretical analysis how such processes can be streamlined, see *ibid*, Chapter 3. Concrete examples: Fedora Guiding Policy. Available at: <https://docs.fedoraproject.org/en-US/council/policy/guiding-policy/>; Fedora Council Charter. Available at: <https://docs.fedoraproject.org/en-US/council/>; Jenkins – Project Governance Document. Available at: <https://www.jenkins.io/project/governance/index.html>.

- Developing workflows for developer requests for incorporating Free Software into their stacks, and requiring third party software suppliers to disclose information about Free Software included in their deliverables¹⁴⁵.
- Creating checklists for license obligations, license and copyright notices. Deploying license handbooks and license compatibility matrix to determine which license are acceptable¹⁴⁶.
- Establishing a baseline compliance by identifying all Free Software components of the project, mapping libraries and dependencies, and performing periodical code scan to check for license compatibility and proprietary element¹⁴⁷.
- Using automation tool for simplifying workflows (i.e. the REUSE linter tool)¹⁴⁸.
- Conducting training for legal/engineering staff to improve intra-team communication on Free Software¹⁴⁹.
- Proposing a contribution policy for handling external contributions from the community¹⁵⁰.

¹⁴⁵ See *ibid*, Chapter 4.

¹⁴⁶ For example, Jilayne Lovejoy, *FINOS Open Source License Compliance Handbook* (1st edn, Fintech Open Source Foundation 2019).

¹⁴⁷ Chapter 5 of Haddad, Ibrahim (2018). *Open Source Compliance in the Enterprise*. 2nd ed. San Francisco: The Linux Foundation. Available at: <https://www.linuxfoundation.org/resources/publications/open-source-compliance-in-the-enterprise>.

¹⁴⁸ See REUSE linter tool documentation. Available at: <https://reuse.readthedocs.io/en/latest/>; For other tools, see: Fossology. Available at: <https://www.fossology.org/about/>. Also see Haddad (n 137), Chapter 6.

¹⁴⁹ For example, internal workshops or intra-team meetings with dedicated agenda on compliance. For a reference on content of such events, see Legal Education Day. Free Software Foundation Europe. Available at: <https://fsfe.org/freesoftware/legal/led.html>.

¹⁵⁰ An external contribution policy is a set of rules that govern the management of third-party FOSS within a project. More precisely, it states the terms on which external contributions are handled and how they are audited and distributed. In some cases, since an external contribution policy is not a strictly legal document, it makes sense to have a legal instrument that you can point contributors to in order to protect your project from code contributions that may infringe third party copyright or patents. Depending on the size and complexity of the contributors network, there are different legal instruments you can use to manage external contributions. For instance, a Developer Certificate of Origin (DCO) sets the incoming license of the external contribution to be the same as the outgoing license of your software project. DCOs reduce the barriers for external developers to contribute, while still requiring them to certify that they are submitting their own work. However, projects may prefer contributors license agreements (CLA) or even copyright assignment agreements, which more heavy-handed manner of ensuring that the software project always has the rights to use the code that has been contributed. See Gabriel Ku Wei Bin and

- Setting up a trademark policy with instructions how third-party Free Software projects can interact with trademarks, if applicable¹⁵¹.

A Free Software compliance program can affect the organization both internally regarding employees and contributors and externally with suppliers, partners, and customers. As seen in this section, the compliance efforts are not related to legal but involves a great deal of community engagement. All facets of an organization can be involved in ensuring proper compliance and contributing to correct Free Software consumption and production. Free Software communities expect respect to the license terms, so making sure they are heard can benefit not only the organization but the whole FOSS environment. The next section illustrates how a harmonized set of best practices can streamline compliance efforts by simplifying how copyright and license information for software projects.

4.5. Standardizing License and Copyright Information for Software Projects: the REUSE Initiative

The multiple competing requirements for communicating information about licenses and copyright may increase the compliance burden on project maintainers, especially for smaller Free Software ¹⁵² (or simply, “REUSE”), is a project founded by the Free Software Foundation Europe in 2017 to raise awareness of the best practices for expressing license and copyright information in Free Software projects. The initiative intends to facilitate management of source codes by making licensing and copyright information more consistent in how it is added to the source code, allowing for automating many of the processes involved. Several different file types have competing and unstandardized methods to store metadata and fields for copyright and license. REUSE defines best practices for declaring copyright and licensing in an

Lucas Lasota, ‘Guidelines for External Contribution Policy - - Next Generation Internet Legal To-Dos’

<<https://download.fsfe.org/NGI0/V2/FSFE%20-%204%20-%20Free%20Software%20Contribution%20Policy.pdf>>.

¹⁵¹ For example, Debian Trademark Policy. Available at: <https://www.debian.org/trademark>; General guidelines for using the Nextcloud marks. Available at: <https://nextcloud.com/trademarks/>; Gabriel Ku Wei Bin and Lucas Lasota, ‘Guidelines for Trademark Policy- Next Generation Internet Legal To-Dos’

<<https://download.fsfe.org/NGI0/V2/FSFE%20-%206%20-%20Trademark%20Policy%20Guidelines.pdf>>.

¹⁵² See the project’s web portal. Available at: <https://reuse.software/>

unambiguous, human- and machine-readable way, so that the information is preserved when the file is copied and reused by third parties. The REUSE specifications aim to facilitate and improve management policies for the digital commons, improving data and metadata communication for individuals, communities, governments, and businesses.

The more external components a software code encompasses, the harder it is to keep an overview of the copyright holders and their licensing choices. Since Free Software licenses are public documents being openly shared - often by millions of users worldwide - their implementation generally does not involve negotiation among the parties. Therefore, proper information regarding the governing license is crucial to avoid legal and security risks. This is especially problematic for Free Software projects, as large public code repositories mean ¹⁵³ licensed repositories. Moreover, license proliferation fragments the requirements for copyright and license notices. Software projects incorporating content elements—as text, images, and videos face an additional layer of complexity with content licensing compliance¹⁵⁴.

How copyright and license information should be displayed depends on copyright law and license requirements¹⁵⁵. Especially important are notices for reciprocal licenses (copyleft), as they require the derivative work to be licensed under the same licensing terms, which directly impacts license compatibility. Although Free Software licenses in general provide information on how the license notices should be applied, the vastly diverse recommendations remain unharmonized. Standardizing these license notices in each source file is beneficial as it helps in an automated analysis of applicable licenses. The REUSE best practices can be advantageous for¹⁵⁶:

- Individual developers – by providing them with a precise and easy-to-implement way for applying correct terms of license and copyright notices.

¹⁵³ Ben Balter, 'Open Source License Usage on GitHub.Com' (*The GitHub Blog*, 10 March 2015) <<https://github.blog/2015-03-09-open-source-license-usage-on-github-com/>> accessed 26 February 2023.

¹⁵⁴ See, for instance, the Creative Commons recommendations for applying a license to creative works. Available at: https://wiki.creativecommons.org/wiki/Marking_your_work_with_a_CC_license Retrieved on 30.06.22.

¹⁵⁵ See Matija Suklje, 'How and Why to Properly Write Copyright Statements in Your Code ... and Probably More than What You Ever Wanted to Know about Them' (2021) <<https://matija.suklje.name/how-and-why-to-properly-write-copyright-statements-in-your-code.>>.

¹⁵⁶ Lucas Lasota, 'REUSE Software - Making Copyright and Licensing Compliance Easier for Everyone', *Proceedings of the Weizenbaum Conference 2022* (2022) <<https://www.weizenbaum-institut.de/events/weizenbaum-conference-2022-practicing-sovereignty-interventions-for-open-digital-futures/>>.

- Digital communities – by improving how data and metadata for software re-usability is communicated.
- Academia – by improving re-usability of software in a safe and clear way in research projects.
- The public sector – by fostering best practices for dealing with license and copyright notices, improving interoperability among agencies, and fostering open government.
- Commercial entities by allowing them to optimize their software bill of materials and simplify development workflow.

Regarding the uptake of REUSE, although it is not possible to know exactly the number of adopters, by February 2023, 1443 software repositories using the REUSE API are successfully implementing and following the best practices. REUSE had been adopted by the Linux Kernel, and several large companies. The specifications are also a central element in the compliance workflow for the European Commission's Next Generation Internet Initiative¹⁵⁷, serving as consortium best practices for software and research projects developing human-centric technologies for the future of the Internet.

4.5.1. REUSE Specifications

REUSE's core specifications are based on Software Package Data Exchange (SPDX)¹⁵⁸, an open standard for communicating software bill of material information, including components, copyrights, licenses and security references. The SPDX project maintains a license list¹⁵⁹, which defines standardized identifiers for a wide spectrum of commonly found licenses and exceptions used in FOSS, data, hardware,

¹⁵⁷ For a detailed overview of the initiative, see the NGI0 Zero website, available at: <https://www.ngi.eu/ngi-projects/ngi-zero/> Retrieved on 30.06.22.

¹⁵⁸ Software Package Data Exchange (SPDX) is an international ISO open standard (ISO/IEC 5962:2021) managed by the Linux Foundation. As Kate Stewart explains, *when the Software Package Data Exchange® (SPDX) project was started in 2010, it was with a simple goal of being able to share summary information about a software package between the creator and consumer. At that time, to comply with the licences in Open Source, you had to find them in the source code. This resulted in hours of working with commercial source scanning tools, and once you had the details, you didn't have a good way of sharing them. After comparing notes and recognising there was a group of managers, lawyers, and developers frustrated by the same problem, we started a grassroots effort to standardise the information that we wanted to share.* See Kate Stewart, 'SPDX and Software Bill of Materials ISO/IEC 5962L 2021' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (2nd edn, Oxford University Press 2022).

¹⁵⁹ The SPDX License List includes a standardized short identifier, the full name, the license text, and a canonical permanent URL for each license and exception. Available at: <https://spdx.org/licenses>.

or documentation. A common language and vocabulary to express security, licensing, and copyright information for products, components, packages, files and code snippets, enable tools to be created and facilitate the introduction of compliance automation.

The REUSE specifications¹⁶⁰ define a standardized method for declaring copyright and licensing for software projects. The goal of the specification is to have unambiguous, human- and machine-readable copyright and licensing information for each individual file in a project. Ideally this information is embedded into every file, so that the information is preserved when the file is copied and reused by third parties. The implementation procedure comprehends of three compliance steps, which will be explained in detail below:

1. Choosing and providing licenses
2. Adding copyright and license information to each file in project's repository
3. Confirming REUSE compliance

4.5.1.1. Choosing and Providing Licenses

Stating the project's license

It is considered best practice for software projects to inform the public of their intended licenses by stating clearly anywhere visible on the project's front page or README file. The actual text of the license need not to be in such places, but the name of the license, as well as a link to the full license text. As seen below, the actual text of the license is located in a specific directory.

Including the license into the project's repository

A project must include a license file for every license under which files in the project are licensed. A LICENSES directory should be created in the project's root which will contain all the licenses used in the project. Each license file must be placed in the LICENSES/ directory in the root of the project. The licenses are identified according to their SPDX identifier of SPDX License List¹⁶¹. For example, the SPDX License Identifier for the GNU GPL v3.0 is "GPL-3.0-only". The license file must be in plain text

¹⁶⁰ This specification implements IETF RFC 2119: Key words for use in RFCs to Indicate Requirement Levels.

¹⁶¹ In case a project uses a custom or modified license, that does not appear in the SPDX License List, such a license should be placed in the file LICENSES/LicenseRef-MyLicense.txt in the repository of the project. Tools using SPDX will nevertheless be able to recognize the custom or modified license.

format. A project must not include license files for licenses under which none of the files in the project are licensed.

4.5.1.2. Adding Copyright and License Information to Each File in Project's Repository

Every single file in the project must have copyright and licensing information associated with it, naming the copyright date, holder, and license¹⁶². This notice should also include a note on where to find the full text of the license. It is possible to associate copyright and licensing information with a file using one of the two ways: comment headers, or through a DEP5 file.

Option 1: Implementing comment headers to files

Each plain text file that can contain comments must contain comments at the top of the file (the comment header) that declare that file's Copyright and Licensing¹⁶³ Information. The comment header must contain one or more SPDX-FileCopyrightText tags, and one or more SPDX-License-Identifier tags. A tag is followed by a colon, followed by a text value, and terminated by a newline.

Assuming that the license is the GPL v3.0, an example of a comment header would be as follows:

¹⁶² It is imperative to always include the name of the copyright holder(s). If the project operates on a CLA, then the name of the responsible legal entity or person has to be mentioned. Attribution helps to avoid orphan works. Attribution is also essential from the perspective that version control typically only records authorship and is unsuitable for the task of recording copyright and hence, the correct copyright information needs to be recorded in each file. The tags can be used multiple times if a project has multiple copyright holders or licenses. However, if a project has an extremely long list of copyright holders and does not want a lengthy copyright notice, it is possible to simply use "Copyright (c) 2013 John Doe et al." as a copyright tag and have a complete list of copyright holders and authors included in a separate AUTHORS.* or CONTRIBUTORS.*file in the project. The syntax should always correspond to each file type. Even if a project wants to put their work in the public domain, it is important to display the correct copyright and license information such as 0BSD for code or CC0-1.0 for non-code in the SPDX format. This should be done in order to allow anyone to contact the correct copyright holder to evade any ambiguity arising from jurisdictional copyright issues. For detailed instructions, see: <https://reuse.software/faq/>.

¹⁶³ As previously mentioned, a summary of the license information of a project can also be recorded in the README file of project and it can also be used to simply redirect the reader to the LICENSES/ directory. The pyproject.toml file of the REUSE tool also declares all the licenses that it uses in the format expected by the Python packaging infrastructure.

```
/*
 * SPDX-FileCopyrightText: 2023 Jane Doe <jane@example.com>
 *
 * SPDX-License-Identifier: GPL-3.0-or-later
 */
```

The `SPDX-FileCopyrightText` tag records the publication years¹⁶⁴ and the copyright holder of the contents of the file. The `SPDX-FileCopyrightText` tag must be followed by a copyright notice¹⁶⁵.

The `SPDX-License-Identifier` tag must be followed by a valid SPDX License Expression, typically just the SPDX License Identifier of the license describing the licensing of the file (example: `SPDX-License-Identifier: GPL-3.0-or-later`). If separate sections of the file are licensed differently, a different `SPDX-License-Identifier` tag must be included for each section¹⁶⁶.

Each file must always contain these two tags in the header. It is possible to use the tags multiple times if the project has multiple copyright holders or licenses.

To add copyright and license information to binary and uncommentable files, which are hard to edit; in order to insert comment headers, one can create the files, for example, `CAT.JPG.LICENSE` and `DOG.JPG.LICENSE`. The same copyright and license information can then be inserted in these files. REUSE maintains that every

¹⁶⁴ Another feature in the copyright tag is the year which could be either the year of initial publication, the year of the latest publication or all years of publications, either as range (e.g., 2017-2019) or as separate entries (e.g., 2017, 2018, 2019). While it is not compulsory to add the year to the copyright tag, recording when the work was first created might just be a useful tool for a patent defence. See also Suklje (n 156).

¹⁶⁵ As a note of caution, attribution is a discretionary right that an author exercises. If the authors wishes to maintain privacy of their identity or have in fact not listed themselves on the project, then it is not obligatory to name them in the project at all. In case the author wants to stay anonymous, the author should simply state the name of the project along with the URL to the project's homepage as a point of contact. When it comes to maintaining anonymity over the project, it is crucial to know that Git stores both author and committer data for each commit and it would be worth checking as to whether or not those records can be linked back to the identity of the anonymous author. For detailed instructions, see: <https://reuse.software/faq/>.

¹⁶⁶ If a project uses code from multiple sources that use licenses texts such as MIT or the BSD, which can be modified to contain custom copyright notices, it can be quite a problematic and tedious task to deal with the license information. This is owing to the reason that both the MIT license and the BSD family of licenses include a clause that requires the redistributor to reproduce the copyright notice and the license text. The pragmatic solution is to put the unmodified license text (i.e., the license text template without any copyright notices) in the `LICENSES/` folder. The copyright notices of the upstream project can then be embedded into the corresponding source code files that were reused, as usual.

single file should contain licensing information, so that even the tiniest of files get a header, are accompanied by a corresponding .license file, or marked in a bulk.

Option 2: Associating Copyright and Licensing information through a DEP5 file

If the project has large directories where including a comment header in each file (or in .license companion files is impossible or undesirable, it is possible to alternatively associate copyright and licensing information with a particular file through a DEP5 file.

The DEP5 file must be named dep5 and stored in the .reuse/ directory in the root of the Project (i.e. .reuse/dep5). The License tag must be followed by a valid SPDX License Expression describing the licensing of the associated files. The Copyright tag must be followed by a copyright notice. An example of a DEP5 file would be as follows:

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: my-project

Upstream-Contact: Jane Doe <jane@example.com>

Source: <https://git.example.com/jane/my-project>

*Files: resources/img/**

Copyright: 2017 Jane Doe <jane@example.com>

License: CC-BY-4.0

*Files: resources/vid/**

Copyright: 2017 Jane Doe <jane@example.com>

2017 John Doe <john@example.com>

License: CC0-1.0

4.5.1.3. Confirming REUSE Compliance

When all files in the repository are marked with their copyright and licensing, a compliance check should be performed¹⁶⁷. To do this, the REUSE linter tool¹⁶⁸ can be used to automate compliance, which provides the following summary:

SUMMARY

Bad licenses: 0

Missing licenses: 0

Unused licenses: 0

Used licenses: CC-BY-4.0, CC0-1.0, GPL-3.0-or-later

Read errors: 0

Files with copyright information: 6 / 6

Files with license information: 6 / 6

Congratulations! Your project is REUSE compliant :-)

4.5.2. REUSE and Related Compliance Initiatives

FOSS license compliance is a vast area populated with a multitude of initiatives and tools to help compliance efforts. REUSE contributes to this effort but does not replace tools and best practices in other places of the compliance chain. In fact, REUSE, which has a community-based approach, supplements several related projects¹⁶⁹, such as

¹⁶⁷ A project may want to exclude some files from the REUSE compliance check for some technical purposes. If the file is a build artifact and a project uses Git, it is important to make sure that the file is covered by a .gitignore file. For insignificant files that are not particularly copyrightable, for example, configuration files, such as .gitignore, one can assign them in the public domain by using the CC0-1.0 license. It is noteworthy that assigning the files to the public domain should only be done by the original author itself. If the file was authored by someone else, it is obligatory to declare their copyright and license in the header. If an entire directory needs to be excluded from REUSE compliance testing, using a DEP5 file would be helpful. If a section containing strings needs to be excluded as it may falsely be detected as copyright or license statements, it can be wrapped within the two comments REUSE-IgnoreStart and REUSE-IgnoreEnd. However, these practices should not be implemented to ignore valid copyright and licensing information by oneself or a third party.

¹⁶⁸ The REUSE linter tool documentation can be found here: <https://reuse.readthedocs.io/en/stable/#lint>.

¹⁶⁹ For an overview of complementary initiatives, see: <https://reuse.software/comparison/>.

ClearlyDefined¹⁷⁰, OpenChain¹⁷¹ and FOSSology¹⁷². The REUSE initiative also offers to software developers a series of resources¹⁷³ for easy engagement and adoptions, an open mailing list for discussion and deliberation, extensive FAQs, and a constantly updated toolkit with compliance tools, API checks, and provision for numerous CI/CD solutions.

After analysing the compliance elements, the next section will handle the issues end-users are having caused by large corporations hampering the ability to install and run Free Software on their devices, as well the policy, legal and legislative reactions to this.

4.6. Free Software Contemporary Issues: the Right to Repair and Regulation Over Corporate Behaviour Towards Digital Devices

The digitization of infrastructures and services comes along with a growing number of electronic devices connected to the Internet. The information and communications sector (ICT) is a driving force in this process. Although digital devices are ubiquitous in the information society, the number of devices on which end-users cannot run Free Software ¹⁷⁴[OBJ]. The consequence is an increased loss of control over users' technology and direct impact on the correlation between Free Software and hardware. The scope of this section is to contextualize the legal aspects of Free Software (many

¹⁷⁰ ClearlyDefined is an Open Source Initiative incubator project. The goals of the project are to collect and display meta and security information about a large number of software and data projects distributed on different package registries. It also motivates developers and curators to extend data about a project's licensing and copyright situation. REUSE in comparison concentrates on fixing the problem at the file level for individual projects. See: <https://clearlydefined.io/about>.

¹⁷¹ The OpenChain Project is focused on building trust in the free software supply chain. OpenChain focuses on making free software license compliance more transparent, predictable, and understandable for participants in the software supply chain. OpenChain recommends REUSE as one component to increase clarity of the licensing and copyright situation but has higher requirements to achieve full conformance. See: <https://www.openchainproject.org/>.

¹⁷² FOSSology is a toolkit for Free Software compliance, stores information in a database, and includes license, copyright and export scanners. It is more complex than REUSE and its helper tool and rather optimized for compliance officers and lawyers. REUSE instead intends to have all licensing and copyright information stored in or next to the source files to safeguard this information when reused elsewhere. See: <https://www.fossology.org/about/>.

¹⁷³ For detailed instructions, see: <https://reuse.software/dev/>.

¹⁷⁴ See Lucas Lasota, 'Why Device Neutrality Is Important for Free Software?' (FOSDEM, Brussels, 2022) <<https://archive.fosdem.org/2022/schedule/event/deviceneutrality/>>.

already addressed in this chapter) can relate to freedoms such as installing and uninstalling software in digital devices and open hardware. For that, an analysis on the latest case law and legislative developments in Europe is carried out, as well the policy responses the Free Software communities have been engaged.

Digital devices are a present reality in all aspects of life. They are used for work, communication, entertainment, and internet access. Such devices are capable machines, allowing users to have access to a high number of features, and perform a large spectrum of tasks. Smartphones, tablets, laptops, and other connected devices can be considered general purpose computers¹⁷⁵, meaning that users can potentially run any compatible software to make full use of the hardware. This potential is fostered by Free Software. Nevertheless, the dominant power of large digital corporations and providers over internet access has sparked policy, regulatory and legal reactions trying to impose accountability on such large enterprises controlling how end-users should use their devices. In recent years several FOSS communities have been involved in public debates and legislative processes in two areas concerning Free Software and devices: the right to repair and economic regulation of corporate behaviour towards devices.

4.6.1. Right to Repair and Free Software

The diverse narratives over the right to repair has gained traction in political and advocacy discourses in the EU in the last years, mostly under broader considerations on sustainability of the digital sector¹⁷⁶. The right to repair relates essentially to the right to obtain information about repairing one's own devices and accessing software and hardware that can be used to do so, which may run contrary to commercial interest of the proprietary companies.

¹⁷⁵ See Erik Albers, 'On the Sustainability of the Free Software' (*Free Software Foundation Europe*, 2021) <<https://fsfe.org/freesoftware/sustainability/sustainability>>. This study explores the definition of software sustainability and shows from there how the inherent characteristics of FOSS enable a sustainability of software as well as their positive impact on the sustainability of IT infrastructures. Software obsolescence is explained and the benefits of using FOSS in saving natural resources by extending hardware usage lifetime and through saving energy consumption of software. The study outlines political demands on digital sustainability.

¹⁷⁶ See Anthony Rosborough, 'Unscrewing the Future: The Right to Repair and the Circumvention of Software TPMs in the EU' (2020) 1 JIPITEC <<https://www.jipitec.eu/issues/jipitec-11-1-2020/5083>>.

In this regard, there have been some recent judicial developments. In the European jurisdiction, in *Top System SA v. Belgian state*¹⁷⁷, a request for a preliminary ruling was brought before the Court to clarify whether Article 5(1) of the Software Directive can be interpreted to allow a lawful customer of a computer program to decompile all or part of the program in order to fix discrepancies impacting the program's operation, including circumstances where the correction consists of disabling a component that is affecting the efficient functioning of the application? If that question is answered affirmatively, must the conditions set forth in Article 6 of the Directive (which allows decompilation), as well as any additional circumstances, also be met? The Court held that a computer program can be decompiled to repair a mistake under Article 5, and that this right is separate from the right to decompile software for interoperability under Article 6.

Similarly, in US, a lawsuit was filed in October 2021 by Software Freedom Conservancy against the TV maker, Vizio, Inc.¹⁷⁸ that uses copylefted software programs in its TVs, on the ground of failure to fulfil the obligations provided under GNU Public License (GPL), namely, the right to access the source code. This claim is central to the right to software repair, as it allows users to exercise the right to copy, share, modify, and reinstall the software on their devices. The lawsuit focuses on the rights of individual consumers as third-party beneficiaries of the GPL to obtain technical information of the source code in the form of a specific performance. This marks as the first case to show individual consumers have rights to the source code as third-party beneficiaries of the GPL. On May 13, 2022, the United States District Court for the Central District of California granted the SFC's motion to have the case remanded back to the Superior Court in Orange County.

On the legislative front, The European Commission in 2020 has published the white paper Circular Economy Action Plan¹⁷⁹ that aims at strengthening consumer protection against premature obsolescence and will work towards establishing a new “right to repair”, including a right to update obsolete software. The proposed plan is

¹⁷⁷ *Top System SA v Belgian State* [2021] ECLI:EU:C:2021:811 (Court of Justice of the European Union).

¹⁷⁸ *Software Freedom Conservancy, Inc v Vizio, Inc* [2022] 8:21-cv-01943-JLS-KES (CD Cal). For a detailed explanation on this court case, see: SFC (2021). *Software Freedom Conservancy files lawsuit against California TV manufacturer Vizio Inc. for GPL violations*. See SFC's press kit, SFC, 'Software Freedom Conservancy Files Lawsuit against California TV Manufacturer Vizio Inc. for GPL Violations' <<https://sfconservancy.org/docs/software-freedom-conservancy-v-vizio-announce-press-kit.pdf>>.

¹⁷⁹ European Commission, 'Circular Economy Action Plan: For a Cleaner and More Competitive Europe' <https://ec.europa.eu/environment/circular-economy/pdf/new_circular_economy_action_plan.pdf>.

inspired by and aims to widen the scope of the Ecodesign Directive of 2009¹⁸⁰, which was implemented through product-specific regulations which aims that devices are designed for energy efficiency and durability, repairability, upgradability, maintenance, reuse and recycling¹⁸¹. In line with these initiatives, there has been an ongoing debate on whether it might be more beneficial to reframe the “right to repair” software to right to¹⁸²:

- Revert to prior versions of a software product – Not every update is an upgrade and often many updates degrade the usability of the software. In this context, it would be highly desirable that a consumer has the ability to revert to prior versions of the software product along with any information necessary to restore the user’s system to the *status quo ante*.
- Refuse updates – As aforementioned, many updates are not desirable for the consumers, and they should have the ability to refuse any automatic or compelling update.
- Receive repairs for a certain period – For consumers interested in sticking to the old versions of the software device, there should be a provision for the company to continue to make and provide repairs instead of opting for sunset repairs.

¹⁸⁰ Directive 2009/125/EC of the European Parliament and of the Council of 21 October 2009 establishing a framework for the setting of ecodesign requirements for energy-related products (recast) (Text with EEA relevance) 2009.

¹⁸¹ The update of the European ecodesign directive includes a set of regulations and directives that together build the framework for the ecodesign criteria of sustainable products within the European Union. With the possibility of more regulations or directives to come and aspects being covered through horizontal ruling, yet we have two regulations directly targeting mobile phones and tablets, which are "Designing mobile phones and tablets to be sustainable – ecodesign" – European Commission. Environmental impact of mobile phones and tablets, 2022. Available at: https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/12797-Designing-mobile-phones-and-tablets-to-be-sustainable-ecodesign_en; and European Commission, Energy labelling of mobile phones and tablets – informing consumers about environmental impact, 2022. Available at: https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/12798-Energy-labelling-of-mobile-phones-and-tablets-informing-consumers-about-environmental-impact_en.

¹⁸² See e.g., European Commission (2021). *Explanatory memorandum for the ecodesign consultation forum: ecodesign and energy labelling—mobile phones, cordless phones and tablets*. ("Software updates of the operating system shall be provided for 5 years, comprising security updates and for at least the first 3 years also functionality updates; such updates shall be provided within a reasonable time after the market introduction of a related release; updates shall not have an adverse effect on device performance, or the user has to have the option to downgrade to the prior version of the operating system"); European Commission (2021). *Preparatory study for the ecodesign and energy labelling working plan 2020-2024*; European Commission (2021). *Preparatory study on mobile phones, smart phones, and tablets: final report*.

In connection to this, the policy concept of “*device neutrality*” and “*reusability of software*” have been pushed forward to deal with unbalanced and non-sustainable corporate control over digital devices. In 2021, the Free Software Foundation Europe kicked-off the Upcycling Android initiative¹⁸³ with the objective to raise awareness and to advocate on the overcoming of software obsolescence and extend the lifespan of users’ hardware with the help of Free Software. The universal right to freely choose operating systems, software and services is crucial for a more sustainable digital society. For this reason, connected services as well as the software on connected devices and applications must offer interoperability and full functionality of a device's initial purpose with the use of open standards¹⁸⁴. To enable interoperability, manufacturers must ensure that any data necessary to run a device's primary function is compatible with and possible to import/export in open standards. Smaller components of a device often require specific drivers, tools, and interfaces to operate. If these are licensed under Free Software licenses, then the reusability of the source code can enable a right to repair for any third parties from professional repair shops to repair cafés to end-users.

Nevertheless, principles of digital sustainability can be incorporated device-oriented policies which reflect the importance of open technologies particularly Free Software to avoid greater impact on the environment. Overcoming hardware obsolescence introduced through software obsolescence would require exchanging the initial operating system with one that is still benefiting from development and support. This is a common solution in the mobile phone and tablet sector. Unfortunately, manufacturers often impose restrictions on the possibility to install third party software and operating systems. The right to install any third-party software on any device would enable users to choose software that helps them to keep running their devices even if the initial manufacturer has decided to stop their support. In fact, the ability to reuse and repurpose resources in a creative and sustainable way, the universal right to install and develop any operating system and software is fundamental¹⁸⁵. It would furthermore enable volatile aftermarkets and enhanced competition regarding re-use

¹⁸³ Upcycling Android website. Available at: <https://fsfe.org/activities/upcyclingandroid/>. The initiative also provided a legal analysis on whether flashing a mobile device (e.g., an Android phone) and replacing its operating system would void statutory warranty in relation to EU consumer law. See: FSFE (2022). Does flashing your device (e.g., an Android phone) and replacing its operating system void your statutory warranty, if you are a consumer located in the European Union? Available at: <https://fsfe.org/activities/upcyclingandroid/is-flashing-legal.en.html>

¹⁸⁴ For an overview on open standards and FOSS, see: FSFE. *Open Standards*. Web page. Available at: <https://fsfe.org/freesoftware/standards/index.en.html>.

¹⁸⁵ Albers (n 176).

of devices. That is why legal, technical, commercial or other obstacles to reuse these devices for any purpose should be discouraged¹⁸⁶.

As seen, diverse Free Software communities have engaged in public debates and legislative processes to translate policy demands into the right for any user and third-party to install any software on any digital equipment and device neutrality. They seek to foster solutions based on Free Software for a critical, long-lasting, and sustainable change and extension of our hardware usage lifetimes, legislation should support broader possibilities of end-user control over devices.

4.7. Regulating Corporate Behaviour Towards Digital Devices in the EU

Although devices such as smartphones, laptops and tablets general purpose computers, device manufacturers, vendors, and internet platforms have been restricting the usage of Free Software by exercising their monopolistic control over end-user equipment. This power is used over key features and components of devices such as operating systems, browsers, and apps stores. Since these elements are essential for the functioning of devices, they constitute a termination monopoly¹⁸⁷, which grants such companies powers in the sense that they could be considered “gatekeepers of gateways” for end-users to access and control software, hardware, and services of their devices. Manufacturers, vendors, and platforms controlling devices may perform a **gatekeeper function** in similar ways to a provider of an internet access connection controlling a gateway to the internet. The more important the device features and components are for end-users, the more entrenched the position of the gatekeeper can become in the digital markets. Monopolized markets are prejudicial to fair competition and consumer welfare. With gatekeepers controlling operating systems, apps stores, browsers, and key online services, end-users are left

¹⁸⁶ The Free Software Foundation Europe has published an open letter asking European legislators to use the update of the ecodesign regulations to establish a universal right to install any software on any device. More than 150 civil society organizations across sectors and companies have already signed it: <https://fsfe.org/activities/upcyclingandroid/openletter.html>.

¹⁸⁷ Competition has been the major regulatory concern for the telecom sector. The entire sector suffered several waves of privatization since 1990, imposing policy makers issues regarding fair competition and consumer welfare. See, for example, chapter 1 of Andrej Savin, *EU Telecommunications Law* (1st edn, Edward Elgar Publishing 2020).

with no or few choices, hindering individual freedoms and self-determination. In general terms, gatekeeper control is achieved by ¹⁸⁸:

- **Restricting usage of Free Software in devices:** Gatekeepers limit or prevent users installing different operating systems, browsers, apps stores, drivers, etc. on their devices. They also impose pre-installed apps on users and make their uninstallation harder or in some times impossible.
- **Locking devices down:** Such companies hinder interoperability, exercise tight control over APIs, and apply proprietary standards, hampering functionalities and blocking access to drivers and hardware.
- **Increasing switching costs:** Gatekeepers keep users in "walled gardens", tie devices to online accounts, bundle apps stores, and hamper data portability, making it harder for users to switch software, devices, and services.

Such monopolistic power of gatekeepers imposes challenges to Free Software and consumer welfare. "*Device neutrality*" is a concept that has been pushed forward by the academic sector since 2013 and received attention in the last years in legislative developments in Europe. It represents ways to disintermediate the power of gatekeepers and re-establish competition in markets and end-user control over devices. In relation to FOSS, end-users should be able to bypass gatekeepers and have the ability to run Free Software on their equipment. Device neutrality's main goal is to resolve the termination monopoly over devices, so users can enjoy software freedom and have access to alternative services and content with their devices.

Re-establishing end-user control over devices and fair competition in digital markets requires safeguarding software freedom in devices, protecting end-users from lock-in, and promoting end-user control over data. In 2022, several components of device neutrality were included in the Digital Markets Act (DMA)¹⁸⁹, the European Union's largest initiative to regulate gatekeepers in digital markets. Although the law contains the principles for making Device Neutrality a reality, its regulations concern only very

¹⁸⁸ This summary is derived from the following studies: J Krämer, 'Device Neutrality: The Missing Link for Fair and Transparent Online Competition?' <https://cerre.eu/wp-content/uploads/2020/05/CERRE_DeviceNeutrality_IssuePaper_March2019.pdf>; J Krämer and R Feasey, 'Device Neutrality: Openness, Non-Discrimination and Transparency on Mobile Devices for General Internet Access' <<https://cerre.eu/publications/mobile-devices-net-neutrality-internet-access/>>.

¹⁸⁹ Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act) 2022.

large platforms¹⁹⁰. The DMA establishes obligations for gatekeepers - “dos” and “don’ts” they must comply with - and prescribes fines and penalties for infringements¹⁹¹.

In relation to Free Software and open hardware, DMA requires gatekeepers to allow side-loading by the users, meaning thereby that users can uninstall any pre-installed software (Art. 6(3)), as well as install and use any third-party apps on their devices (Art. 6 (4)). Furthermore, DMA includes several provisions to enhance the protection against vendor lock-in. Gatekeepers cannot prohibit app developers to set different prices and conditions for their apps in alternative stores (e.g., F-Droid) (Art. 5(3)). Gatekeepers cannot make software developers and end-users register or sign in for a service or app as a condition for access to their or another gatekeeper's service or app (Art. 5(7)). Besides, they should allow end-users to access and use third-party apps, content, subscriptions, features, or other functionalities (Art. 5(5)). Finally, Gatekeepers cannot restrict the ability of end-users to switch between and subscribe to different apps and services that are accessed via the operating system or apps store (Art. 6(6)). Gatekeepers are also required to provide the same hardware and software features accessed or controlled via the operating system to third-party developers (Art. 6(7)). This includes data interoperability (also called real-time data portability (Art. 6(9)) and interoperability among messaging apps (Art. 7). These provisions provide for a stepping ground for the Free Software ecosystem to develop in the digital markets.

4.8. Conclusion

Free Software licenses caused a revolution in how software is consumed and produced. Such licenses were designed to streamline the reusability of software in ways the movement developed in accordance with principles of software freedom. Licenses can enable specific outcomes, such as keeping obligations as simple as possible or enforcing copyleft to protect the sharing of improvements. In any case, compliance efforts are necessary to respect license terms. Such efforts involve not only legal-oriented actions, but also involve governance elements such as organizational processes and community management. The larger the community or the organization working collectively with, the more complex a compliance program may become. Diverse initiatives have tried to ease the compliance burden. REUSE is

¹⁹⁰ See DMA Art. 2 and Art. 3. for subject matter and scope of the law. *ibid* 2, 3.

¹⁹¹ For a detailed analysis of the DMA and FOSS, see FSFE (2022). Digital Markets Act and Free Software, FSFE. Available at: <https://fsfe.org/activities/dma/dma.html>.

one of such projects offering easy-to-implement best practices for copyright and license information. With the greater adoption of Free Software by consumers and producers, issues of dominant corporate behaviour in digital markets have sparked legislative and policy reaction of decision makers and civil society in Europe. The concept of “device neutrality” has been pushed forward as a catalyst of the right to repair and other initiatives in the field of digital sustainability.

5. Primer on Open Hardware Licensing

The open source hardware definition adopted by OSHWA is concerned with tangible artifacts, such as machines, devices, physical things whose design has been released to the public with modification rights. This definition is based on the open source software definition, so it is perhaps not surprising that OSH licensing has followed closely the licensing principles applicable to OSS.

The following sub-sections provide an overview of the challenges to applying reciprocal and permissive licenses to hardware. We lay emphasis on the major open source licenses, such as the CERN Open Hardware License v1.0 and 2.0, the Solderpad license, and TAPR. We finally discuss the different development models for OSH and why they are different to the models known in OSS.

5.1. Scope of “Open Hardware” and Licensing Issues

The term open hardware covers a wide range of tangible and intangible goods, such as mechanical items, electronic items, or aesthetic items. Typically, the distribution of an open hardware project involves providing a link to the original design files for the hardware on the product, packaging and documentation. It also entails product labelling with version numbers and release dates, so that there is a correspondence between the physical object and the respective version of the design files. It is also essential that open source parts are clearly delineated from non-open source parts and that there is a logo on the parts that are covered by design files released under an open hardware license. The OSHWA provides detailed information on best practices for users.¹⁹²

As previously explained, this means that there is an array of diverse IP rights that may apply to open hardware. Furthermore, the distinction between tangible and intangible in hardware is less clear compared to software. For example, in electronics we distinguish between a printed circuit board, which is a physical item, and components like field programmable gate arrays (FPGA), whose operation is determined by

¹⁹² See more on the open source hardware practices here: <https://www.oshwa.org/sharing-best-practices/>

bitstream, which is akin to software¹⁹³. This implies that the legal basis for the use, modification and distribution is much less understood compared to software.

Furthermore, as Katz explains, the economics of hardware are different. The 'instantiation of any piece of hardware is liable to require non-trivial effort', so the 'differential in cost between re-engineering a design in a way which does not impinge on any IPRs, and using an existing open hardware design ... is likely to be smaller than the equivalent scenario in software'¹⁹⁴.

There is also the so-called boundary problem which 'applies in terms of the components of which a design is composed' ¹⁹⁵. The problem is linked with the tangible nature of hardware and the breadth of the range of classes and categories of item to which the term 'hardware' applies. As we shall see later, this problem is addressed by the reciprocal variants of the CERN OHL v2 by exempting from the requirement to provide the complete design documentation any component which qualifies as 'Available Components'.

Finally, despite their widespread use, open source software (e.g., GPL) and open culture or content licenses (e.g., Creative Commons) are ill-suited for open hardware because the terminology they use does not map well to other subject matter ¹⁹⁶. For example, it is not clear what the equivalent of a computer program's object code would be for a hardware design. It may sound intuitive to argue that pieces like the bitstream are akin to computer programs. However, the definitions used in copyright law and many of these licences may have conceptual difficulties to accommodate such a broad definition.

The OSHWA has identified the following licenses are relevant to hardware:

- Reciprocal licenses
 - Creative Commons Attribution, Share-Alike (CC-BY SA)
 - GNU GPL
 - Hardware-specific licenses: TAPR OHL, CERN OHL

¹⁹³ Andrew Katz, 'Everything Open' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) 525–526

<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>.

¹⁹⁴ Katz, 'Open Hardware' (n 1) 499.

¹⁹⁵ *ibid* 500.

¹⁹⁶ *ibid* 504.

- Non-reciprocal (permissive) licenses
 - Free BSD license (BSD-2 Clause)
 - MIT license (MIT)
 - Creative Commons Attribution (CC-BY-3.0)
 - Hardware-specific licenses: Solderpad Hardware License

The following section discusses how hardware-specific open source licenses operate and what types of rights over what subject matter they license.

5.2. Reciprocal Licenses and Open Hardware

As Katz explains, applying reciprocal licensing obligations to hardware designs is more difficult because hardware depends on a broader range of IP rights than copyright ¹⁹⁷. He has argued that 'any license which tries to echo the GPL by requiring the distribution of hardware to be accompanied by the source will necessarily be limited in its effectiveness by virtue of the extensive opportunities for making use of the underlying design without having to rely on the license' ¹⁹⁸. Indeed, 'using a piece of hardware, or transferring a piece of hardware from one person to another does not potentially contravene any intellectual property rights, and therefore does not require any license on which copyleft-type requirements can impinge' ¹⁹⁹.

Differently from copyleft, open hardware is faced with barriers unknown to developers of open source software, such as²⁰⁰:

- Replication could be costly
- Equipment required to replicate hardware is typically more expensive than the cost of a simple computer and compiler/IDE
- The test/fix/test cycle for hardware is much longer than that of software
- Creating hardware requires available physical spaces that also has to meet certain environmental conditions
- Transporting hardware is much more challenging than transferring bits

¹⁹⁷ *ibid* 496.

¹⁹⁸ Andrew Katz, 'Towards a Functional Licence for Open Hardware' (2012) 4 *Journal of Open Law, Technology & Society* 41, 45.

¹⁹⁹ *ibid*.

²⁰⁰ *ibid* 53.

- Collaborating effectively on hardware at a distance is a challenge that does not exist for software
- There is limited availability of free and open source tools for CAD, CAM etc.
- Testing hardware could be prohibitively expensive and it is also usually linked with the much more complex regulatory regime around hardware certification.

Even so, some major open source hardware licenses seek to apply a form of reciprocity, notably the CERN Open Hardware License.

5.2.1. CERN Open Hardware License

This license was developed by CERN to promote collaboration among hardware designers and support the freedom to use, study, modify, share and distribute hardware designs, and products based on these designs. CERN is a custodian of this license and is responsible for releasing new versions and variants of the license. The current version of the license is version 2. The license comes in three variants ²⁰¹, two reciprocal and one permissive:

- CERN-OHL-S (strongly reciprocal)
- CERN-OHL-W (weakly reciprocal)
- CERN-OHL-P (permissive)

CERN OHL focuses on design documentation, so the user rights under this family of licenses are granted once the user performs an act that would impinge on the exclusive rights in the design documentation. Three definitions are particularly important in this context:

- ‘*Source*’, defined as ‘information such as design materials or digital code which can be applied to Make or test a Product or to prepare a Product for use, Conveyance or sale, regardless of its medium or how it is expressed. It may include Notices’²⁰².
- ‘*Covered Source*’, defined as ‘Source that is explicitly made available under this License’²⁰³.

²⁰¹ Javier Serrano, ‘CERN OHL Version 2’ <<https://ohwr.org/project/cehnohl/wikis/Documents/CERN-OHL-version-2>> accessed 20 February 2023.

²⁰² CERN OHL Version 2 – Strongly Reciprocal, section 1.3.

²⁰³ CERN OHL Version 2 – Strongly Reciprocal, section 1.4.

- ‘*Product*’, defined as ‘any device, component, work or physical object, whether in finished or intermediate form, arising from the use, application or processing of Covered Source’²⁰⁴.
- ‘*Complete Source*’, defined as ‘the set of all Source necessary to Make a Product, in the preferred form for making modifications, including necessary installation and interfacing information both for the Product, and for any included Available Components. If the format is proprietary, it must also be made available in a format (if the proprietary tool can create it) which is viewable with a tool available to potential licensees and licensed under a license approved by the Free Software Foundation or the Open Source Initiative. Complete Source need not include the Source of any Available Component, provided that You include in the Complete Source sufficient information to enable a recipient to Make or source and use the Available Component to Make the Product’.²⁰⁵
- ‘*Available Component*’, defined by CERN-OHL-S as ‘any part, sub-assembly, library or code which:
 - a) is licensed to You as Complete Source under a Compatible License; or
 - b) is available, at the time a Product or the Source containing it is first Conveyed to You and any other prospective licensees
 - i. as a physical part with sufficient rights and information (including any configuration and programming files and information about its characteristics and interfaces) to enable it either to be Made itself, or to be sourced and used to Make the Product; or
 - ii. as part of the normal distribution of a tool used to design or Make the Product’²⁰⁶
- ‘*Available Component*’, defined by CERN-OHL-W as ‘any part, sub-assembly, library or code which:
 - a) is licensed to You as Complete Source under a Compatible License; or
 - b) is available, at the time a Product or the Source containing it is first Conveyed, to You and any other prospective licensees

²⁰⁴ CERN OHL Version 2 – Strongly Reciprocal, section 1.5.

²⁰⁵ CERN OHL Version 2 – Strongly Reciprocal, section 1.8.

²⁰⁶ CERN OHL Version 2 – Strongly Reciprocal, section 1.7.

- i. with sufficient rights and information (including any configuration and programming files and information about its characteristics and interfaces) to enable it either to be Made itself, or to be sourced and used to Make the Product; or
- ii. as part of the normal distribution of a tool used to design or Make the Product'²⁰⁷.

As highlighted in the user guide to the CERN-OHL-S v2, authorship or ownership of the design must be clear and undisputed for the license to be operational²⁰⁸.

One of the distinguishing features of the CERN-OHL family of licenses is the introduction of the 'Available Components' category. Electronic designs often include commonly available elements (e.g., resistors, transistors, capacitors) which are accompanied by corresponding data sheets. In these cases, there is no need to provide any source. However, this may not be the case for specialist components where information needs to be provided enabling it to be made²⁰⁹.

Although the CERN-OHL has been released in two reciprocal variants, the distinction between them is subtle. Andrew Katz provides the following example of open source chip designs which are often released as source code written in a hardware description language, such as Verilog ²¹⁰. This code describes the interconnection of components, like the logic gates. Other components can be imported as libraries; these would normally qualify as 'available components'. The CERN-OHL-S allows available components to be provided without the complete source where they are physical components and come with readily available interfacing information. For libraries provided as code, as in the example above, this exemption would not apply. This means that a designer would not be able to combine code licensed under the CERN-OHL-S with other components to make an HDL design if the complete source for these other components is not available ²¹¹.

This position is relaxed in the CERN-OHL-W variant which provides that non-physical components (e.g., code libraries and silicon chip designs) can also qualify as 'available components'. Similarly, if they are available and there is interfacing information, the library code does not have to be released. This means that CERN-

²⁰⁷ CERN OHL Version 2 – Strongly Reciprocal, section 1.7.

²⁰⁸ CERN, 'CERN Open Hardware Licence Version 2 - Strongly Reciprocal - User Guide' 1
<https://ohwr.org/project/cernohl/wikis/uploads/cf37727497ca2b5295a7ab83a40fcf5a/cern_ohl_s_v2_user_guide.pdf> accessed 20 February 2023.

²⁰⁹ Katz, 'Open Hardware' (n 1) 505–506.

²¹⁰ *ibid* 506.

²¹¹ *ibid*.

OHL-W can be combined with components released under a different license (incl. proprietary), so long as the CERN-OHL-W licensed components can be integrated with the rest of the design using the documented interface ²¹².

The ability to combine CERN-OHL-W with components licensed under different licenses is crucial for OSH developers. Many of the software tools used to design silicon chips are proprietary and contain proprietary libraries. The scarcity of open source toolchains seems to be a significant problem precisely because of the code incorporated by the toolchain into the output and its effect from a legal point of view²¹³. For example, there is no uniform position on the question whether the bitstream should be treated as a computer program under copyright law, or as something else²¹⁴.

Additionally, CERN-OHL-W has adopted patent license and retaliation clause. The authors of the CERN-OHL v2 recognized designers may want to have the hardware design and the software (e.g., firmware) licensed under the same license. The CERN-OHL were therefore submitted to the OSI for official approval as OSS licenses, which was granted by the OSI in 2021²¹⁵.

5.2.2. TAPR

Another popular choice for an OSH license is the TAPR Open Hardware License (TAPR-OHL). In the words of its creator, John Ackermann, the license 'provides a framework for hardware projects that is similar to the one used for Open Source software' ²¹⁶. Importantly, he recognizes that '[t]his isn't as straight-forward as it seems because legal concepts that work well for software (such as copyright and copyleft) don't neatly fit when dealing with hardware products and the documentation used to create them' ²¹⁷.

Like the CERN-OHL, the focus of the TAPR-OHL is on design documentation. One distinctive feature of the license is that it is 'not primarily a copyright license'. The preamble clarifies this further:

²¹² *ibid* 507.

²¹³ Andrew Katz, 'A Survey of Open Processor Core Licensing' (2019) 10 *Journal of Open Law, Technology & Society* 21, 30.

²¹⁴ *ibid*.

²¹⁵ Katz, 'Open Hardware' (n 1) 507.

²¹⁶ John Ackermann, 'TAPR Open Hardware License' (*Tomorrow's Ham Radio Technology Today*, 2007) <<https://tapr.org/the-tapr-open-hardware-license/>> accessed 20 February 2023.

²¹⁷ *ibid*.

'While copyright protects documentation from unauthorized copying, modification, and distribution, it has little to do with your right to make, distribute, or use a product based on that documentation. For better or worse, patents play a significant role in those activities. Although it does not prohibit anyone from patenting inventions embodied in an Open Hardware design, and of course cannot prevent a third party from enforcing their patent rights, those who benefit from an OHL design may not bring lawsuits claiming that design infringes their patents or other intellectual property'

The license does not cover software, firmware or code loaded into programmable devices, referring developers to copyright-oriented licenses for this type of subject matter.

TAPR-OHL defines 'documentation' as including schematic diagrams; circuit or circuit board layouts, including Gerber and other data files used for manufacture; mechanical drawings, including CAD, CAM, and other data files used for manufacture; flow charts and descriptive text; and other explanatory material²¹⁸. Similarly, 'products' are defined as subject matter based in whole or in part on the documentation and covering specifically: circuit boards, mechanical assemblies, and other physical parts and components; assembled or partially assembled units (including components and subassemblies); and parts and components combined into kits intended for assembly by others²¹⁹.

TAPR-OHL is essentially a copyleft license which, however, acts as a contract. It achieves this effect by binding those who rely on the license to release any modifications to the design under the same license. However, as noted by Andrew Katz, this may be a problem in common law jurisdictions because of the requirement for consideration of contract. This problem is addressed by granting a patent non-assert, which is claimed to be an effective consideration ²²⁰. Specifically, section 2.4 of the TAPR-OHL provides that:

These grants of immunity are a material part of this Agreement, and form a portion of the consideration given by each party to the other. If any court judgment or legal agreement prevents you from granting the immunity required by this Section, your rights under this Agreement will terminate and you may no longer use, copy, modify or distribute the Documentation, or make, have made, or distribute Products²²¹.

²¹⁸ TAPR-OHL, section 1.2.

²¹⁹ TAPR-OHL, section 1.3.

²²⁰ Katz, 'Open Hardware' (n 1) 504–505.

²²¹ TAPR-OHL, section 2.4.

It should be noted that none of the clauses of the TAPR have been tested in a court of law, so we do not know how they may be interpreted and whether patent non-assert would be accepted as an effective consideration. Nevertheless, the most significant concern remains the problem of applying copyleft, which is a tweaked version of the copyright license bargain, to hardware²²².

5.3. Non-reciprocal (Permissive) Licenses and Open Hardware

5.3.1. CERN-OHL-P

Like many permissive licenses, the permissive variant of the CERN-OHL v2 is more lenient compared to the copyleft versions. Accordingly, CERN-OHL-P does not mention 'Available Components'. It provides that the licensor may copy and convey verbatim or modified copies of covered source, in any medium, provided they retain all notices and, for modified covered source, also add notice stating that it has been modified and how. Provided these notices are kept and a copy of the license is provided, the licensor may convey both covered and modified covered source under different license terms, including proprietary.²²³

5.3.2. Solderpad Hardware License

The Solderpad Hardware License was developed by Andrew Katz as an alternative to copyleft-oriented TAPR and CERN-OHL v1, which did not include a permissive variant. It is a non-copyleft form of license based on Apache 2.0 and adapted to the needs and problems of open hardware developers²²⁴.

The Apache 2.0 license was identified as the best candidate for a license that would work for hardware with minimal modifications due to its popularity and patent and trademark clauses. The main change in the Solderpad license concerns the extension of licensed rights to cover not only copyright but also other relevant rights, such as

²²² Katz, 'Towards a Functional Licence for Open Hardware' (n 198) 42.

²²³ CERN OHL Version 2 – Permissive, sections 3.1-3.4.

²²⁴ Katz, 'Towards a Functional Licence for Open Hardware' (n 198) 42.

design rights, semiconductor topography rights and database rights ²²⁵. The Solderpad license also amends the 'Source' definition to include net lists, board layouts and CAD files:²²⁶

“Source” form shall mean the preferred form for making modifications, including but not limited to source code, net lists, board layouts, CAD files, documentation source, and configuration files.

The definition of 'Object' includes intermediate forms, such as bytecode, bitstream, artwork and semiconductor topographies²²⁷:

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form or the application of a Source form to physical material, including but not limited to compiled object code, generated documentation, the instantiation of a hardware design or physical object or material and conversions to other media types, including intermediate forms such as bytecodes, FPGA bitstreams, moulds, artwork and semiconductor topographies (mask works).

The scope of 'Derivative Works' has also been clarified as not including any work which physically connects or interoperates with the interfaces of the Work²²⁸:

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship or design. For the purposes of this License, Derivative Works shall not include works that remain reversibly separable from, or merely link (or bind by name) or physically connect to or interoperate with the Work and Derivative Works thereof.

Finally, the license grant, in its latest version, provides that²²⁹:

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable license under the Rights to reproduce, prepare

²²⁵ *ibid* 54.

²²⁶ Solderpad Hardware License v2.1, section 1.

²²⁷ *ibid*.

²²⁸ *ibid*.

²²⁹ Solderpad Hardware License, section 2.

Derivative Works of, make, adapt, repair, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form and do anything in relation to the Work as if the Rights did not exist.

Importantly, the Solderpad Hardware License must be read in conjunction with the Apache License²³⁰. Section 1 modifies definitions and terminology in the Apache License and Section 2 below replaces Section 2 of the Apache License²³¹.

²³⁰ Solderpad Hardware License, preamble.

²³¹ *ibid.*

6. Principles and Concurrent Phenomena of Open Data Licensing

6.1. Data, Intellectual Property and Intellectual Property Rights

6.1.1. What Exactly is Data?

When we talk about *data*, we talk about a wide range of *material*: raw data, quantitative data, qualitative data, aggregated data, data records, datasets, databases, data products and data services – to name some. When we talk about *intellectual property*, we talk about wide range of *creations with intellectual input*: literature, paintings, images, designs, software, and inventions – to name some. And when we talk about *intellectual property rights*, we talk about wide range of *forms of legal protection* granted for the intellectual property: copyrights, patents, trademarks, and database rights – to name some. The distinction between these three elements is crucial in forming understanding how the licensing of open data operates in real-life-cases.

The origin of the term of data is in Latin and its singular form meant a fact given or granted, (thing) given, a fact given as the basis for calculation in mathematical problems, or numerical facts collected for future reference.²³²

According to Merriam-Webster Dictionary data means: “1. factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation; 2. information in digital form that can be transmitted or processed; 3. information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful.”²³³ From this definition it becomes clear that data is something that is meant to be further processed in order to refine and develop it further.

²³² ‘Datum’ <<https://www.etymonline.com/word/datum>> accessed 26 February 2023.

²³³ ‘Definition of Data’ <<https://www.merriam-webster.com/dictionary/data>> accessed 26 February 2023.

Cambridge Dictionary defines a dataset as “a collection of separate sets of information that is treated as a single unit by a computer”²³⁴ and database as “a large amount of information stored in a computer system in such a way that it can be easily looked at or changed”²³⁵. These definitions highlight the digital nature of data and its processing.

One of the more recent and specific definition of data is identified in connection with the Montreal Data License, which reads as follows: “By data, we refer to the information being made available. The format and layout of such information is referred to as a database or dataset, whatever way it may be organized. Where applicable, the data may be separated into different segments into underlying data or metadata in the form of data tags and other structural information. Such data may be collected and harnessed from different sources, or made available from a single-source. Data can be basic collated information (e.g., a range of measurements such as temperature, location) or be formed of more complex information (e.g., pictures, maps).”²³⁶

Finally, terms like data product and data-as-a-service emphasize the end-user aspects relating to data, the need to build products or provide services that are based on data.

6.1.2. Data as Intellectual Property

None of the above definitions emphasize the element of intellectual property in relation to data. Can data be regarded as intellectual property? One attempt to define whether data is above the threshold of intellectual input that is required to grant it intellectual property rights, is the regulation of rights related to databases in the European Database Directive²³⁷. The Database Directive, Article 3 identifies copyright protection for databases and Article 7 the sui generis database right. For fulfilling the threshold of intellectual input required for copyright protection under Article 3, said directive requires that the database “by reason of the selection or arrangement of their contents, constitute the author's own intellectual creation” whereas for the lower level sui generis database right under Article 7 it is sufficient that “there has been

²³⁴ ‘Dataset’ <<https://dictionary.cambridge.org/dictionary/english/dataset>> accessed 26 February 2023.

²³⁵ ‘Database’ <<https://dictionary.cambridge.org/dictionary/english/database>> accessed 26 February 2023.

²³⁶ Misha Benjamin and others, ‘Towards Standardization of Data Licenses: The Montreal Data License’ (arXiv, 20 March 2019) <<http://arxiv.org/abs/1903.12262>> accessed 2 November 2022.

²³⁷ Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases 1996 (OJ L).

qualitatively and/or quantitatively a substantial investment in either the obtaining, verification or presentation of the contents to prevent extraction and/or re-utilization of the whole or of a substantial part, evaluated qualitatively and/or quantitatively, of the contents of that database”. The field of application of the sui generis database right has caused a lot of uncertainty. The latest attempt to clarify the borderline of protected and non-protected database can be seen in the Data Act proposal, Article 35, in which the Commission proposes to clarify that “the sui generis right provided for in Article 7 of Directive 96/9/EC does not apply to databases containing data obtained from or generated by the use of a product or a related service”²³⁸.

Linking data to the copyright regime has been criticized due to the nature of data lacking relevant intellectual input. Data is not directly comparable to work of art and it is questionable to what extent data is an intellectual creation. For instance, Heather Meeker has stated it bluntly: “Data is a special case, and for data, the only good license is no license—or more accurately, there’s no need for a copyright license in the first place.”²³⁹

However, it should also be noted that datasets can also contain copyrighted material. This is the case for many publicly available datasets²⁴⁰. Taking into account the European sui generis database right, there are three basic types of combinations of databases and creative works²⁴¹. In addition, the data and databases may be further processed, for instance, by AI and ML technology.

An additional point to bear in mind is that apart from intellectual property rights, data or data embedded in datasets or databases, may be subject to other rights, such as rights relating to personal data protection, image rights (e.g., name, photo and likeness, etc.), publicity rights (e.g., misappropriation of name, photo, voice etc.).²⁴²

This leads to a system in which different types of data with different levels of protection and rights are embedded in datasets and databases, then further processed to further

²³⁸ European Commission, Proposal for a Regulation of the European Parliament and of the Council on harmonised rules on fair access to and use of data (Data Act) 2022.

²³⁹ Heather Meeker, ‘Beyond Open Data: The Only Good License Is No License’ [2022] *PLI Chronicle: Insights and Perspectives for the Legal Community* <[<https://plus.pli.edu/Details/Details?fq=id:\(352066-ATL2\)>](https://plus.pli.edu/Details/Details?fq=id:(352066-ATL2))>.

²⁴⁰ Gopi Krishnan Rajbahadur and others, ‘Can I Use This Publicly Available Dataset to Build Commercial AI Software? -- A Case Study on Publicly Available Image Datasets’ (arXiv, 11 April 2022) <<http://arxiv.org/abs/2111.02374>> accessed 2 November 2022.

²⁴¹ Simone Aliprandi, ‘Open Licensing and Databases’ (2012) 4 *Journal of Open Law, Technology & Society* 5.

²⁴² Benjamin and others (n 236).

data, datasets, databases and even copyrighted work. The system becomes very difficult to understand and manage.

On a general level there are two ways to effectively handle the usage rights of these combinations of rights relating to data: (i) waiver of rights and (ii) specific licensing of the rights.²⁴³ This will be further investigated under Section 6.3 through the investigation of different types of open data licenses.

6.1.3. Data as a Service

When looking at data from the perspective of intellectual input, the amount of human effort or intellectual component, seems to arise somewhere on the level of database, data products or data services. The higher one looks on the steps of productization or servitization, the less weight is given for possible intellectual property rights as such and the more to the value that the end user gets from the data.

For quite some time now, the trend in digital economy has been shifting from ownership-based economy and physical goods towards licenses to intellectual property and fully digital content. And lately, there has been an increasing trend towards providing digital content as a service. This highlights a major shift from the perspective of intellectual property. If the content – or data – is provided based on subscription as a service, the usage rights are granted with additional limitations, like for the duration of agreed period and the content may change dynamically. In such cases the user gets a usage right to the content, however that is not necessarily based on any underlying intellectual property right but is instead a mere contractual commitment. Service providers can build services, irrespective whether or not there is an underlying intellectual property right. With this trend, also data becomes the object of trade as a service and is no longer commercialized through licensing of databases or similar intellectual property.²⁴⁴

6.2. Types of Open Data Licenses

²⁴³ Aliprandi (n 241).

²⁴⁴ Hans Graux, 'Data Sharing as a Service: Will Data Services Remove Intellectual Property Rights from the Picture, and at What Cost?' (Publications Office of the European Union 2021) <<https://data.europa.eu/doi/10.2830/815190>> accessed 20 February 2023.

6.2.1. Traditional Categorization of Open Data Licenses

Open data licenses can be divided into three major categories:

1. Public domain licenses, containing e.g.
 - CC01.0
 - Open Data Commons Public Domain Dedication and License
2. Permissive licenses (Non-reciprocal), containing e.g.
 - CC-BY
 - Community Data License CDLA Permissive
 - Open government license (for public sector information)
3. Copyleft licenses (Reciprocal), containing e.g.
 - CCBY-SA
 - ODbL

6.2.2. Compatibility

The Open Knowledge Foundation has a list of licenses as conformant with the definition of 'open'.²⁴⁵

Such licenses conform to the Open Definition and are:

- Reusable:
 - Not specific to an organization or jurisdiction.
- Compatible:
 - Must be compatible with at least one of GPL-3.0+, CC-BY-SA-4.0, and ODbL-1.0.
 - Permissive/attribution-only licenses must be compatible with all 3 of the aforementioned licenses, and at least one of Apache-2.0, CC-BY-4.0, and ODC-BY-1.0.

²⁴⁵ Open Knowledge Foundation, 'Conformant Licenses - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge' <<https://opendefinition.org/licenses/>> accessed 26 February 2023.

- Current:
 - Widely used and generally considered best practice by a broad spectrum of projects and actors within the domains of applicability of the license.

Table 19. List of Licenses Conformant with the Definition of ‘Open’

License (SPDX)	Domain	By	SA	Comments
Creative Commons CCZero (CC0-1.0)	Content, Data	N	N	Dedicate to the Public Domain (all rights waived)
Open Data Commons Public Domain Dedication and License (PDDL-1.0)	Data	N	N	Dedicate to the Public Domain (all rights waived)
Creative Commons Attribution 4.0 (CC-BY-4.0)	Content, Data	Y	N	
Open Data Commons Attribution License (ODC-By-1.0)	Data	Y	N	Attribution for data(bases)
Creative Commons Attribution Share-Alike 4.0 (CC-BY-SA-4.0)	Content, Data	Y	Y	
Open Data Commons Open Database License (ODbL-1.0)	Data	Y	Y	Attribution-ShareAlike for data(bases)

6.2.3. Problems of Using Traditional Licenses for Data

Using traditional licenses for the distribution of data contains several problems. These traditional licenses stem mostly from open source software licensing or copyright licensing. Therefore, due to the differences in the IP rights basis of software, content and data, the traditional licenses contain irrelevant terms or terms that are difficult or even impossible to adhere to with respect to data. In addition, such license terms cause challenges with respect to data governance and compliance. These collisions with regard to license terms highlight the need for more robust and coherent management of data within single organizations and among organizations when opening and sharing data.

For instance, Heather Meeker has identified several problems in open data licensing that do not exist in the same manner for instance with FOSS licensing: Mere notice requirements create obstacles as datasets and records are not used and build in the same way as software packages. Data interacts with software differently than software interacts with other software, this is apparent especially in the case of AI. Resolving global challenges in a reliable manner would require a different kind of flow of data compared to the siloed and restricted access taking place at the moment. The

balancing act between revelation of the intellectual property to the public (such as a work of art or an invention) and granting of intellectual property rights protection (such as copyright to a work of art or a patent to protect an invention) does not work with regard to data that often lacks the element of intellectual input as a lot of data is created automatically.²⁴⁶

6.2.4. Novel Approaches

During recent years, there have been several initiatives aimed to tackle the problems generated by the use of FOSS licenses in open data licensing. These initiatives approach the open data licensing from the perspective of certain technologies and focus on how the data is actually used within such technologies. We will address two of these that fall into the scope of the identified emerging technology fields identified in the ZOOM project plan. The first one is the Montreal Data License, or actually a tool to generate such licenses, addressing the use of data in the field of artificial intelligence (AI) and machine learning (ML). The second one is the Cryptographic Autonomy License addressing the use of data in connection with open source software relating to blockchain technology.

6.2.4.1. Montreal Data License tool

The goal of the Montreal Data License is to build a more standardized, transparent, predictable framework with ambiguous language, a framework that would resemble FOSS licensing. It aims to clarify concepts and tools for the use of data in AI and ML and foster market creation for data and level playing field for all market participants. AI and ML require good quality data, i.e., in addition to large volumes, data that is organized, labelled and tagged. The problems in current open data licenses include among other issues lack of identifying different types of specific use; the ambiguity in the terms commercial, non-commercial and research use; lack of standard terminology and standardization; and share-alike requirements that pose difficulties in connection with use of data in the AI and ML technologies. The core of the Montreal Data License tool is in (i) the identification of the core elements in the interaction of data and models and (ii) the distinction and specification of the use of the data itself and the use of data in connection with the models. The core definitions include for instance data, model (untrained or trained), representation and output. The

²⁴⁶ Meeker, 'Beyond Open Data: The Only Good License Is No License' (n 239).

specification of use for data itself contains terms access, labelling, distribute, represent; and the specification of use of the data in conjunction with models contains terms benchmark, research, publish, internal use, output commercialization, model commercialization.²⁴⁷ Already from the above, one can clearly see the value that such context-specific licenses give for making the environment of open data licensing more clear and predictable. This may lead to more standardized practices affecting the data economy as a whole.

6.2.4.2. Cryptographic Autonomy License

Cryptographic Autonomy License aims to protect the end users of crypto software and distributed peer-to-peer (P2P) software by protecting the users' control of their cryptographic keys. These keys can be described being both user data and functional as code. The license does this by permitting the public performance (one element of copyright) only if the end-user's autonomy of their private cryptographic keys. In case these keys would be compromised, the user's control over their data and software would be compromised.²⁴⁸ Again, this kind of context-specific formulation of the license and use of the elements afforded by copyright demonstrates ways to bring clarity on how data and copyrights interact with each other.

6.3. Concurrent Topics Regarding Open Data

6.3.1. Trend towards services and ecosystems

One concurrent trend in the digital business is the trend towards providing services based on subscription fees instead of single purchases. Examples of this are cloud services (such as iCloud, Google Drive) and streaming of video and audio content (such as Netflix, Spotify).²⁴⁹ This trend affects directly how data is processed and

²⁴⁷ Benjamin and others (n 236).

²⁴⁸ Arthur Brock, 'Understanding the Cryptographic Autonomy License' (*Holochain*, 22 February 2019) <<https://medium.com/holochain/understanding-the-cryptographic-autonomy-license-172ac920966d>> accessed 26 February 2023.

²⁴⁹ Graux (n 244).

consumed in business as well, we have seen for instance services like Azure and AWS, which are widely used.

The trend of providing data as a service has a profound effect on how data is shared and licensed between entities. As referred to above in Section 7.1.1.3, data is no longer seen as a strange cousin of intellectual property, but instead from the perspective of the end-user as something that adds value to the end-user. Data is seen as something that forms part of the services, not as a single licensable item. This trend can be seen in the latest context-specific open data licensing terms referred to above in Section 7.3.4, in contrast with the more traditional forms of licensing stemming from licensing data embedded under specific legal forms of intellectual property rights.

The servitization trend opens new possibilities for building business upon data and highlights the need for novel business design and business models. It also highlights the need for data sharing between entities and governance mechanisms for such data sharing, for instance through data ecosystems. The concept of open data ecosystems, focusing thus purely on open data, is currently an emerging concept.²⁵⁰

Provision of data as a service and emergence of data ecosystems demonstrate data as a resource with non-rival nature and focuses on how data can flow between entities.

6.3.2. Regulatory Trends

One of the forerunners in opening data come from actors in the field of public sector data, governmental data and open science. European regulation covering these fields has for some time paved the way towards data as a service. For example Open Data Directive (2019) continued from the basis laid down by the Public Sector Information Directive (2013) emphasizing dynamic data sharing and issues like data access and provision of application programming interfaces (APIs). For instance, APIs are required for the high value data sets like geospatial data, environmental data and mobility data. Typically, these datasets contain data that is not covered by copyrights or database rights. Again, this emphasizes the trend of servitization of data instead of relying on mere licensing of the underlying intellectual property rights. Similar regulations can also be found from other sectors like banking, energy and automotive industry.²⁵¹

²⁵⁰ Per Runeson, Thomas Olsson and Johan Linåker, 'Open Data Ecosystems -- an Empirical Investigation into an Emerging Industry Collaboration Concept' (arXiv, 3 September 2021) <<http://arxiv.org/abs/2109.01378>> accessed 2 November 2022.

²⁵¹ Graux (n 244).

From the regulatory perspective, data is currently in the spotlight. The so-called Big Five regulations (Data Markets Act, Data Services Act, Data Governance Act, Data Act and AI Act) are in different phases of implementation. The themes pushed forward with these acts include among other issues fair competition in the digital markets, obligations for platforms and gatekeepers, obligations for service provider and data intermediaries, re-use of public sector data, data altruism, IoT-data, data spaces and transparency and risk assessment of AI technology. These will profoundly affect the boundaries on how to conduct data-based business and give incentives for novel types of business in the European data economy.

The Big Five regulations will also give a boost for open data by affecting the business environment in which data flows in the continuum of the data spectrum, from closed to open, and different forms of data sharing in between. At the same time, the systemic challenges embedded in the data sharing and licensing between entities, and on a more general level within the ecosystems and communities, surface. These challenges contain issues like interoperability, APIs, access control, metadata, value generation, regulatory aspects, contractual issues, licensing, ecosystem governance and data governance. Due to the multidisciplinary nature of these challenges, they can be tackled only by addressing them from several perspectives: e.g., from technological, business, legal, operational and political.

6.3.3. Role of Data Management, Data Governance and Compliance

The systemic challenges introduced by the new data economy require focusing on the data management and compliance. There are case studies that show potential risks of license violations in using publicly available datasets, especially when used for commercial purposes.²⁵²

There are several useful resources available for building compliance for open data. However only a few focus specifically on data. Most of these focus on open source compliance for FOSS. Despite this, we can use these compliance frameworks for learnings to be taken into account also for open data compliance processes (e.g., Open Source Compliance in the Enterprise - Handbook)²⁵³. However, at all times it should be kept in mind that essential differences exist between software and data,

²⁵² Rajbahadur and others (n 240).

²⁵³ Haddad (n 137).

and therefore OSS license compliance cannot be used as such to data license compliance.²⁵⁴

Additionally, there is some guidance that focuses on certain aspects of open data management (e.g. Open Data Goldbook for Data Managers and Data Holders for publishing open data²⁵⁵ and governance relating to license compliance for publicly available datasets²⁵⁶).

However, when using the above-mentioned best practices, one should keep in mind the changing landscape of European data economy. Data management, processes and compliance for tomorrow's business based on open data – solely or partly – requires combining of useful elements from these existing governance and compliance frameworks and best practices. In addition, to build effective data management for open data, several layers need to be addressed: internal data management within entities, bilateral data sharing between the entities and multilateral ecosystem governance.

6.4. Building Business on Open Data

ODI (Open Data Institute) has visualized the continuum of data from the perspective of its openness as a data spectrum. In its one end there is closed data that is accessible internally within a single entity. On its other end there is the open data licensed under open license terms to anyone. In between these extremes there is shared data that can vary from named access, to group-based access and public access. Step by step the level of openness increases²⁵⁷. The data spectrum gives perspective to the business use of data. Business can be created based on different types of data and different types of data can be used simultaneously.

To operate in such a business environment, a company needs to understand different regimes of data and business opportunities afforded by these regimes. In addition, using open data or opening own data for use in business requires assessment of risks involved. To be able to make a proper risk assessment, a company should be able to

²⁵⁴ Rajbahadur and others (n 240).

²⁵⁵ 'Relaunch: Open Data Goldbook for Data Managers and Data Holders | Data.Europa.Eu' <<https://data.europa.eu/en/news-events/news/relaunch-open-data-goldbook-data-managers-and-data-holders>> accessed 30 November 2022.

²⁵⁶ Rajbahadur and others (n 240).

²⁵⁷ 'The Data Spectrum' <<https://www.theodi.org/about-the-odi/the-data-spectrum/>> accessed 26 February 2023.

assess the level of freedom it has for the use, re-use and distribution of data. Based on current licensing practices (licensing terms used and compliance and decision making processes within the company) this is a laborious, if not impossible task to complete.

This uncertainty embedded in the sharing of data between entities acts as a hindrance for the use of open or shared data in business. Uncertainty for the use of data stems from several sources. There may be lack of understanding that data has a different nature than copyrighted work or other intellectual property has. Misunderstanding on the nature of data leads to unclarity on how the data should be passed on to others to reuse, in other terms how the data should be licensed. This can be seen clearly from the open source licenses that cover data. In addition to covering data, some of them cover works that are protected by copyrights, such as software, and some of them databases that are protected by database rights. The uncertainty is multiplied when data providers use license terms designed solely for open source software licensing when distributing data.²⁵⁸

In addition, the data used by businesses is formed as collections from different sources of data. This does not fit to any single, clearly distinguished category of data or form of intellectual property. Instead, it comes in heterogenous forms and combinations. Uncertainty creates waste of resources as the actors need to investigate ambiguous licensing terminology, it causes market and incentive imbalances, contractual breaches and risks to privacy.²⁵⁹ Additional complexity is introduced in case the data contains personal data protected by regulations like the European GDPR. In these cases, balancing of the risks and harms relating to opening data and value that can be generated by the use of such data becomes essential.²⁶⁰ Combining and mixing different types of data and compliance requirements from regulations and licensing schemes emphasize the need for coherent management of data.

The trend towards data as a service gives another aspect to the provision of data. Provision of data becomes less linked to the rigid licensing of distinct categories of intellectual property rights and puts the aspect of the end-user in the forefront. Instead of licensing particular data or selling a single data set, business is created by selling

²⁵⁸ Meeker, 'Beyond Open Data: The Only Good License Is No License' (n 239).

²⁵⁹ Benjamin and others (n 236).

²⁶⁰ Publications Office of the European Union., Capgemini Invent., and European Data Portal., *Open Data and Privacy*. (Publications Office 2020) <<https://data.europa.eu/doi/10.2830/532195>> accessed 30 November 2022.

good quality of data or real-time access to data. In other words, selling “immediacy and accuracy” by adding value generated by human effort on top of data.²⁶¹

²⁶¹ Meeker, ‘Beyond Open Data: The Only Good License Is No License’ (n 239).

7. Definition of ZOOM Legal Cases

The scarcity of legal cases has motivated the need to define legal cases which should drive our work in the following deliverables. ZOOM focuses on matching business models and licensing strategies for knowledge generators in four main areas of new and emerging technologies. These technologies include artificial intelligence, quantum computing, industrial robotics and blockchain.

From the vantage point of intellectual property law, as applied to open technologies, we have specified four specific legal cases. These cases are inspired from both ongoing litigation (e.g., the GitHub Copilot case) and traditional open source licensing issues. They represent a spectrum of interrelated legal issues concerning open source software, hardware and data.

The relationship and overlaps between software, hardware and data have already been recognized in scholarship. Thus, as Blind et al. observe regarding open source software and open source hardware:

“OSS and OSH occupy overlapping domains, and the third body of open innovation, open data, interacts with both of them (...) Open technologies in both software and hardware frequently rely on data sources to operate. This is true of machine learning algorithms, which can require large amounts of training data, and web services which rely on services such as mapping data. In fact, there is an increasing number of data sources available through APIs which are available on an open licensing model (for example, Openstreetmap is available under the ODbL and is relied on by a vast number of services.”²⁶²

Furthermore, they also notice that there is an even more considerable overlap between different categories of subject matter comes with additional challenges which are not presently reflected in the IP legal framework:

“[There is an] overlap between what counts as software and hardware, on the one hand, and data on the other. For example, software may rely on lookup tables associating ISO country codes with the country names, and hardware designs may include a netlist which is essentially a database listing the interconnections between components in an electronic circuit (...) [D]ata has the additional challenge that it may contain personal information which creates tension between the rights

²⁶² Blind, Knut and others (n 29) 335.

afforded to data subjects over their personal data and the rights granted by open data licensing. (...) Any policy consideration or recommendation which relates to OSSH should also bear in mind the interrelationship between OSSH and open data”²⁶³

The following legal cases are developed from the perspective of this interrelatedness of different subject matter. In the second deliverable under this work package, we aim to identify how licenses applicable to different subject matter interact with each other. We shall explore these issues from the perspective of license compatibility across subject matter and license choice depending on the results from the business model studies in Work Package 2.

7.1. Machine Learning Programming Assistance Software

A company wants to build an open AI programming assistance software for program synthesis using a large language model for code. Relevant questions include:

- Which license elements are important in the machine learning pipeline regarding:
 - Corpus?
 - Trainer?
 - Model?
 - Output?
- Is the resulting model an adaptation or translation of a “work”?
- What are the available avenues for protection and licensing?

7.2. Quantum Programming Language

A company has designed an open programming language for quantum computing applications. Another company wants to commission customizations to the language for a specific project. Relevant questions include:

²⁶³ ibid 336.

- Are programming languages protected by copyright?
- What are the available avenues for protection and licensing?

7.3. Industrial Robotics Device Design

A research institute has developed a new robotic device for industrial applications and has released its design files under an open (hardware) license. A private company decides to produce and commercialize a product based on this design. Relevant questions include:

- Is the legal protection for inventions, utility models or industrial designs sufficient to ensure the sustainability of such business models?
- What licenses are available for designs released as open hardware and have they been used successfully?
- What are the available avenues for protection and licensing?

7.4. Blockchain

A private company has launched a blockchain-based IP marketplace which enables rightsholders to transact with their portfolios of IP assets, particularly patents. The platform's growing popularity has turned it into a valuable source of information on patent transactions. Over time, patent offices have recognized the platform and its dataset as an important resource that could increase the transparency of patent transactions and contribute to the public disclosure of the patent system. Relevant questions include:

- How to reconcile the commercial interests of platform operators and rightsholders with the public interest of disclosure pursued by patent offices?
- What are the available avenues for protection and licensing of the platform's database?

8. Conclusion

This deliverable introduced concepts that are crucial for a proper understanding of the legal issues pertaining to free and open source software, open (source) hardware and open data. It provided an exposition on the principles of free and open source software licensing, open hardware licensing and open data licensing. The main concern with the application of the principles of open source software licensing to other types of subject matter relates to the different underlying IP rights. Unlike software, which is protected primarily as a literary work by copyright law, hardware and data are much more fluid in terms of the IP regimes that apply to them. To a certain extent, this is probably also why we do not find many examples (or any, indeed) of successful business models that apply the principles of open hardware licensing or open data licensing. To this end, the second deliverable of this work package, is going to focus on how licensing works when products and services combine different subject matter. The identified four legal cases will be used as a point of departure in this analysis as most of them depend on more than one type of subject matter. Finally, the deliverable that follows up to this work will focus on how the choice of open (source) components plays a role in the choice of business model and potential sources of revenue generation.

Bibliography

- Albers E, 'On the Sustainability of the Free Software' (*Free Software Foundation Europe*, 2021) <<https://fsfe.org/freesoftware/sustainability/sustainability>>
- Alevropoulos P, 'The Threat of Software Patents Persists' (*Free Software Foundation*, 2021) <<https://endsoftwarepatents.org/2021/08/the-threat-of-software-patents-persists/>>
- Aliprandi S, 'Open Licensing and Databases' (2012) 4 *Journal of Open Law, Technology & Society* 5
- Aliprandi S and Piana C, *A Complete Manual with a Theoretical Introduction and Practical Suggestions* (2012)
- Ballhausen M, 'Copyright Enforcement' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>
- Balter B, 'Open Source License Usage on GitHub.Com' (*The GitHub Blog*, 10 March 2015) <<https://github.blog/2015-03-09-open-source-license-usage-on-github-com/>> accessed 26 February 2023
- Benjamin M and others, 'Towards Standardization of Data Licenses: The Montreal Data License' (arXiv, 20 March 2019) <<http://arxiv.org/abs/1903.12262>> accessed 2 November 2022
- Benkler Y, *The Wealth of Networks: How Social Production Transforms Markets and Freedom* (Yale University Press 2006)
<https://www.benkler.org/Benkler_Wealth_Of_Networks.pdf>
- Blind, Knut and others, 'The Impact of Open Source Software and Hardware on Technological Independence, Competitiveness and Innovation in the EU Economy' (Publications Office of the European Union 2021) Final Study Report
<<https://ec.europa.eu/newsroom/dae/redirection/document/79021>>
- Böhm M, 'Economics of Open Source' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>

Brock A, 'Understanding the Cryptographic Autonomy License' (*Holochain*, 22 February 2019) <<https://medium.com/holochain/understanding-the-cryptographic-autonomy-license-172ac920966d>> accessed 26 February 2023

Brock A, 'Business and Revenue Models and Commercial Agreements' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>

CERN, 'CERN Open Hardware Licence Version 2 - Strongly Reciprocal - User Guide' <https://ohwr.org/project/cernohl/wikis/uploads/cf37727497ca2b5295a7ab83a40fc5a/cern_ohl_s_v2_user_guide.pdf> accessed 20 February 2023

Cheng M and Choudhary M, 'Chapter 12 - Export Control' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (2nd edn, Oxford University Press 2022)

'Database' <<https://dictionary.cambridge.org/dictionary/english/database>> accessed 26 February 2023

'Dataset' <<https://dictionary.cambridge.org/dictionary/english/dataset>> accessed 26 February 2023

'Datum' <<https://www.etymonline.com/word/datum>> accessed 26 February 2023

De Martin JC and Mélanie R, *The Digital Public Domain Foundations for an Open Culture* (Open Book Publishers 2012)

'Definition of Data' <<https://www.merriam-webster.com/dictionary/data>> accessed 26 February 2023

DiBona C, Ockman S and Stone M (eds), *Open Sources: Voices from the Open Source Revolution* (O'Reilly 1999)

Dizon M, 'Free and Open Source Software Communities, Democracy and ICT Law and Policy' (2010) 18 International Journal of Law and Information Technology

EFF, 'Software Patents and Rise of Patent Trolls' (*Electronic Frontier Foundation*, 2013) <<https://www.eff.org/de/deeplinks/2013/02/deep-dive-software-patents-and-rise-patent-trolls>>

European Commission, 'Circular Economy Action Plan: For a Cleaner and More Competitive Europe' <https://ec.europa.eu/environment/circular-economy/pdf/new_circular_economy_action_plan.pdf>

Ezrachi A and Stucke M, *How Big-Tech Barons Smash Innovation and How to Strike Back* (Harper Collins Publishers 2022)

Free Software Foundation, 'What Is Free Software?' (*GNU Project*)

<<https://www.gnu.org/philosophy/free-sw.html>> accessed 19 February 2023

Gardler R and Walli S, 'Evolving Perspective on Community and Governance' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (2nd edn, Oxford University Press 2022)

GitHub, 'Licensing a Repository' (*GitHub Docs*) <<https://ghdocs-prod.azurewebsites.net/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/licensing-a-repository>> accessed 19 February 2023

Graux H, 'Data Sharing as a Service: Will Data Services Remove Intellectual Property Rights from the Picture, and at What Cost?' (Publications Office of the European Union 2021) <<https://data.europa.eu/doi/10.2830/815190>> accessed 20 February 2023

Haddad I, 'Understanding the Open Source Development Model'

—, *Open Source Compliance in the Enterprise* (2nd edn, The Linux Foundation 2018) <<https://www.linuxfoundation.org/resources/publications/open-source-compliance-in-the-enterprise>>

Haddad I and Warner B, 'Establishing an Open Source Software Strategy: Key Considerations and Tactical Recommendations'

Haff G, *How Open Source Ate Software: Understand the Open Source Movement and So Much More* (Springer 2021)

Haunss S, *Conflicts in the Knowledge Society: The Contentious Politics of Intellectual Property* (Cambridge University Press 2013)

Javier Serrano, 'CERN OHL Version 2'

<<https://ohwr.org/project/cernohl/wikis/Documents/CERN-OHL-version-2>> accessed 20 February 2023

John Ackermann, 'TAPR Open Hardware License' (*Tomorrow's Ham Radio Technology Today*, 2007) <<https://tapr.org/the-tapr-open-hardware-license/>> accessed 20 February 2023

Katz A, 'Towards a Functional Licence for Open Hardware' (2012) 4 *Journal of Open Law, Technology & Society* 41

—, 'A Survey of Open Processor Core Licensing' (2019) 10 *Journal of Open Law, Technology & Society* 21

—, 'Everything Open' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford

University Press 2022) <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>

——, ‘Open Hardware’ in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022) <<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>

Kirschner M, ‘Democracy Requires Free Software’ (*Free Software Foundation Europe*, 2010) <<https://fsfe.org/freesoftware/democracy.en.html>>

Kostakis V, ‘The Advent of Open Source Democracy and Wikipolitics: Challenges, Threats, and Opportunities for Democratic Discourse.’ (2011) 7 *Human Technology*

Krämer J, ‘Device Neutrality: The Missing Link for Fair and Transparent Online Competition?’ <https://cerre.eu/wp-content/uploads/2020/05/CERRE_DeviceNeutrality_IssuePaper_March2019.pdf>

Krämer J and Feasey R, ‘Device Neutrality: Openness, Non-Discrimination and Transparency on Mobile Devices for General Internet Access’ <<https://cerre.eu/publications/mobile-devices-net-neutrality-internet-access/>>

Ku Wei Bin G and others, ‘EU Export Control Regime and Free Software’ <<https://download.fsfe.org/NGI0/V2/FSFE%20-%208%20-%20EU%20Export%20Control%20Regime%20and%20Free%20Software.pdf>>

Ku Wei Bin G and Lasota L, ‘Guidelines for External Contribution Policy - - Next Generation Internet Legal To-Dos’ <<https://download.fsfe.org/NGI0/V2/FSFE%20-%204%20-%20Free%20Software%20Contribution%20Policy.pdf>>

——, ‘Guidelines for Trademark Policy- Next Generation Internet Legal To-Dos’ <<https://download.fsfe.org/NGI0/V2/FSFE%20-%206%20-%20Trademark%20Policy%20Guidelines.pdf>>

Lasota L, ‘Why Device Neutrality Is Important for Free Software?’ (FOSDEM, Brussels, 2022) <<https://archive.fosdem.org/2022/schedule/event/deviceneutrality/>>

——, ‘REUSE Software - Making Copyright and Licensing Compliance Easier for Everyone’, *Proceedings of the Weizenbaum Conference 2022* (2022) <<https://www.weizenbaum-institut.de/events/weizenbaum-conference-2022-practicing-sovereignty-interventions-for-open-digital-futures/>>

Leith P, *Software and Patents in Europe* (1st edn, Cambridge University Press 2007) <<https://www.cambridge.org/core/product/identifier/9780511495267/type/book>> accessed 26 February 2023

Linux Foundation, 'Enabling Easier Collaboration on Open Data for AI and ML with CDLA-Permissive-2.0' (2021) <<https://www.linuxfoundation.org/press/press-release/enabling-easier-collaboration-on-open-data-for-ai-and-ml-with-cdla-permissive-2-0>>

Lovejoy J, *FINOS Open Source License Compliance Handbook* (1st edn, Fintech Open Source Foundation 2019)

Meeker H, *Open (Source) for Business: A Practical Guide to Open Source Software Licensing* (Third Edition, Independently published 2020)

——, *Open (Source) for Business: A Practical Guide to Open Source Software Licensing - Third Edition* (Third Edition, Independently published 2020)

——, 'Open Source Software Licensing Basics for Corporate Users' (2020) <<https://www.youtube.com/watch?v=gF4b1TA5Q5w>>

——, 'Beyond Open Data: The Only Good License Is No License' [2022] *PLI Chronicle: Insights and Perspectives for the Legal Community* <[https://plus.pli.edu/Details/Details?fq=id:\(352066-ATL2\)](https://plus.pli.edu/Details/Details?fq=id:(352066-ATL2))>

Neary D, '6 Pivotal Moments in Open Source History' (*opensource.com*, 2018) <<https://opensource.com/article/18/2/pivotal-moments-history-open-source>>

Open Knowledge Foundation, 'Open Definition v2.1' (*Open Definition*, November 2015) <<http://opendefinition.org/od/2.1/en/>> accessed 19 February 2023

——, 'Conformant Licenses - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge' <<https://opendefinition.org/licenses/>> accessed 26 February 2023

——, 'Open Knowledge Foundation Mission' (*Open Knowledge Foundation*) <<https://okfn.org>> accessed 19 February 2023

Open Source Hardware Association, 'Open Source Hardware (OSHW) Definition 1.0' (*Open Source Hardware Association*, 26 May 2012) <<https://www.oshwa.org/definition/>> accessed 20 February 2023

Open Source Initiative, 'The Open Source Definition' (*Open Source Initiative*, 7 July 2006) <<https://opensource.org/osd/>> accessed 19 February 2023

Patent Absurdity (Directed by Luca Lucarini, Free Software Foundation 2010) <<https://patentabsurdity.com/about.html>>

Peterson S, 'Why so Little Love for the Patent Grant in the MIT License?' (*opensource.com*, 23 March 2018) <<https://opensource.com/article/18/3/patent-grant-mit-license>> accessed 26 February 2023

Phipps S, 'Permissive and Copyleft Are Not Antonyms' (*Open Source Initiative Website*, 2017) <<https://opensource.org/node/875>>

Publications Office of the European Union., Capgemini Invent., and European Data Portal., *Open Data and Privacy*. (Publications Office 2020) <<https://data.europa.eu/doi/10.2830/532195>> accessed 30 November 2022

Rajbahadur GK and others, 'Can I Use This Publicly Available Dataset to Build Commercial AI Software? -- A Case Study on Publicly Available Image Datasets' (arXiv, 11 April 2022) <<http://arxiv.org/abs/2111.02374>> accessed 2 November 2022

Raymond E, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (O'Reilly 1999)

'Relaunch: Open Data Goldbook for Data Managers and Data Holders | Data.Europa.Eu' <<https://data.europa.eu/en/news-events/news/relaunch-open-data-goldbook-data-managers-and-data-holders>> accessed 30 November 2022

Rosborough A, 'Unscrewing the Future: The Right to Repair and the Circumvention of Software TPMs in the EU' (2020) 1 JIPITEC <<https://www.jipitec.eu/issues/jipitec-11-1-2020/5083>>

Rub GA, 'Copyright Survives: Rethinking The Copyright-Contract Conflict' (2017) 103 Virginia Law Review 1141

Runeson P, Olsson T and Linåker J, 'Open Data Ecosystems -- an Empirical Investigation into an Emerging Industry Collaboration Concept' (arXiv, 3 September 2021) <<http://arxiv.org/abs/2109.01378>> accessed 2 November 2022

Savin A, *EU Telecommunications Law* (1st edn, Edward Elgar Publishing 2020)

Schießle B, 'Free Software, Open Source, FOSS, FLOSS - Same but Different - FSFE' (*FSFE - Free Software Foundation Europe*, 8 December 2012) <<https://fsfe.org/freesoftware/comparison.html>> accessed 26 February 2023

Schmitz P-E, 'Licence Compatibility, Permissivity, Reciprocity and Interoperability' (*European Commission - JoinUp - EUPL*, 2023) <<https://joinup.ec.europa.eu/collection/eupl/licence-compatibility-permissivity-reciprocity-and-interoperability>>

SFC, 'Software Freedom Conservancy Files Lawsuit against California TV Manufacturer Vizio Inc. for GPL Violations' <<https://sfconservancy.org/docs/software-freedom-conservancy-v-vizio-announce-press-kit.pdf>>

Shemtov N, *Beyond the Code: Protection of Non-Textual Features of Software* (Oxford University Press 2017)

<<https://oxford.universitypressscholarship.com/10.1093/oso/9780198716792.001.0001/oso-9780198716792>> accessed 13 May 2022

Stewart K, 'SPDX and Software Bill of Materials ISO/IEC 5962L 2021' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (2nd edn, Oxford University Press 2022)

Suklje M, 'How and Why to Properly Write Copyright Statements in Your Code ... and Probably More than What You Ever Wanted to Know about Them' (2021)
<<https://matija.suklje.name/how-and-why-to-properly-write-copyright-statements-in-your-code.>>

'The Data Spectrum' <<https://www.theodi.org/about-the-odi/the-data-spectrum/>>
accessed 26 February 2023

'Urteile - IfrOSS.Org' (ifrOSS, 2023) <<https://ifross.github.io/ifrOSS/Cases.html>>
accessed 24 February 2023

Vezina B and Pearson S, 'Should CC-Licensed Content Be Used to Train AI? It Depends.' (*Creative Commons*, 2021)
<<https://creativecommons.org/2021/03/04/should-cc-licensed-content-be-used-to-train-ai-it-depends/>>

Walden I, 'Open Source as Philosophy, Methodology, and Commerce: Using Law with Attitude' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (New Edition, Second Edition, New Edition, Second Edition, Oxford University Press 2022)
<<https://global.oup.com/academic/product/open-source-law-policy-and-practice-9780198862345?cc=gb&lang=en&>>

Weber S, *The Success of Open Source* (Harvard Univ Press 2005)

Wheeler D, 'The Free-Libre / Open Source Software (FLOSS) License Slide'
<<https://dwheeler.com/essays/floss-license-slide.html>>

WIPO, *Understanding Copyright and Related Rights* (2nd edn, WIPO 2016)
<https://www.wipo.int/edocs/pubdocs/en/wipo_pub_909_2016.pdf>

SAS Institute Inc v World Programming Ltd [2012] ECLI:EU:C:2012:259 (Court of Justice of the European Union)

Software Freedom Conservancy, Inc v Vizio, Inc [2022] 8:21-cv-01943-JLS-KES (CD Cal)

Top System SA v Belgian State [2021] ECLI:EU:C:2021:811 (Court of Justice of the European Union)

Welte / Sitecom Deutschland GmbH [2004] Az 21 O 6123/04 (LG München)

Directive 2009/125/EC of the European Parliament and of the Council of 21 October 2009 establishing a framework for the setting of ecodesign requirements for energy-related products (recast) (Text with EEA relevance) 2009

European Commission, Proposal for a Regulation of the European Parliament and of the Council on harmonised rules on fair access to and use of data (Data Act) 2022

Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act) 2022

Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases 1996 (OJ L)

Regulation (EU) 2021/821 of the European Parliament and of the Council of 20 May 2021 setting up a Union regime for the control of exports, brokering, technical assistance, transit and transfer of dual-use items (recast) 2021 (OJ L 206, 1162021)

'From the Public Sector Information (PSI) Directive to the Open Data Directive | Shaping Europe's Digital Future' (15 February 2023) <<https://digital-strategy.ec.europa.eu/en/policies/psi-open-data>> accessed 14 March 2023

'Open Government Data - OECD' <<https://www.oecd.org/gov/digital-government/open-government-data.htm>> accessed 14 March 2023

'Open Science | UNESCO' <<https://www.unesco.org/en/open-science>> accessed 14 March 2023

'EUR-Lex - 32019L1024 - EN - EUR-Lex' <<https://eur-lex.europa.eu/eli/dir/2019/1024/oj>> accessed 14 March 2023

'EUR-Lex - 32003L0098 - EN - EUR-Lex' <<https://eur-lex.europa.eu/eli/dir/2003/98/oj>> accessed 14 March 2023

'EUR-Lex - 32013L0037 - EN - EUR-Lex' <<https://eur-lex.europa.eu/eli/dir/2013/37/oj>> accessed 14 March 2023

'Mission' <<https://okfn.org>> accessed 14 March 2023

'The Open Definition - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge' <<http://opendefinition.org/>> accessed 14 March 2023

'Open Definition 2.1 - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge' <<http://opendefinition.org/od/2.1/en/>> accessed 14 March 2023

'What Is Open?' <<https://okfn.org>> accessed 14 March 2023

Appendix

This appendix represents a collection of summaries of legal cases involving open source software. As of yet, there are no reports of any legal cases before a court of law, either solved or pending, concerning open source hardware or open data.

This collection is based on the list of cases compiled by the German Institute for Legal Issues of Free and Open Source Software (Institut für Rechtsfragen der Freien und Open Source Software).²⁶⁴

Case name	Country	Court	Key elements
LG Mannheim, comparison of 27.12.2021, Az. 7 O 2/21	Germany	District Court of Mannheim	<ul style="list-style-type: none"> – Settlement applies to the Linux kernel (www.kernel.org) and all programs and program libraries published by the netfilter/iptables project on the website wwwv.netfilter.org (the "Software") – Parties mutually undertake not to enforce in the future the infringement of their copyrights (including copyrights in joint authorship and/or adaptations) in the Software and/or the infringement of the license terms of the GNU General Public License in connection with the Software without the prior consent of the majority of the then active members of the Netfilter Core Team. – Settlement has effect in favour of third parties: licensees of the Software who find themselves exposed to the assertion of claims arising from the infringement of copyrights (including copyrights in joint authorship and/or adaptations) of a party may oppose to that party the lack of consent.
OLG Karlsruhe, Urteil v. 13.11.2020, Az. 6 U 60/20	Germany	Karlsruhe Higher Regional Court	<ul style="list-style-type: none"> – Even if the editor of a WordPress theme is required by the GPLv2 terms to make his or her adaptation freely available to the general public under the GPLv2 as open source (so-called copyleft), such an obligation on the part of the editor vis-à-vis the authors of the original program does not provide third parties with any publication or exploitation rights to this adaptation.
OLG Hamburg, Urteil v.	Germany	Hanseatic Higher Regional	<ul style="list-style-type: none"> – A modification of software may also give rise to an editor's copyright. Parts of computer programs may also be eligible for protection if they themselves meet

²⁶⁴ 'Urteile - IfrOSS.Org' (ifrOSS, 2023) <<https://ifrOSS.github.io/ifrOSS/Cases.html>> accessed 24 February 2023.

28.02.2019, Az. 5 U 146/16		Court Hamburg 5 th Civil Senate	<p>the requirements for copyright protection. (Rn.90) (Rn.91)</p> <ul style="list-style-type: none"> – The author of the arrangement may demand the cessation of the use of another computer program that uses the parts protected for the author of the processor within the meaning of § 69c No. 2 UrhG (reworking) in conjunction with § 23 UrhG (processing) in a non-free manner. The application for an injunction to be formulated procedurally is already sufficiently precise if the version of the program containing the infringement is described with sufficient precision. (para. 93) – When asserting copyright in software, it must be sufficiently substantiated and proven which parts of the software (here: Linux) have been reworked by the plaintiff software developer and in what way, to what extent these modifications meet the requirements for authorship pursuant to § 69c No. 2 sentence 2 in conjunction with § 3 UrhG and that precisely the reworked program parts justifying protection for him have in turn been taken over and used by the alleged infringer (possibly further reworked). (Rn.103)
LG Hamburg, Urteil v. 20.11.2017; Az. 308 O 343/15	Germany	Hamburg Regional Court 8 th Civil Chamber	<ul style="list-style-type: none"> – An act of infringement gives rise to the presumption of a risk of repetition not only for the identical form of infringement, but also for essentially similar acts of infringement (see, inter alia, BGH, Order of 3 April 2014, I ZB 42/11, GRUR 2014, 706); this may also apply to claims arising from declarations to cease and desist (see Cologne Higher Regional Court, Judgment of 24 May 2017, 6 U 161/16).(para.139) – A cease-and-desist declaration may be interpreted to the effect that acts of infringement based on a single decision is to be regarded as only one act of infringement (margin no. 157). – If a cease-and-desist creditor has the right under the cease-and-desist declaration submitted by the debtor to determine the amount of the contractual penalty in the event of a breach of the cease-and-desist obligation at its equitable discretion in accordance with the so-called new Hamburg custom, the determination of the penalty amount made by the creditor is only binding if it is in accordance with equity (cf. inter alia OLG Karlsruhe, judgment of 18 December 2015, 4 U 191/14).(marginal no.175)
OLG Hamm, Urteil v. 13.06.2017, Az. 4 U 72/16	Germany	Hamm Higher Regional Court	<ul style="list-style-type: none"> – The plaintiff is not entitled to claim damages - either under Section 97 (2) UrhG or on the basis of another basis for a claim. It is not apparent that the plaintiff could have suffered any damage as a result of the defendant's conduct of which it complained.

			<ul style="list-style-type: none"> – The plaintiff wants to determine the damages according to the principles of license analogy - which are the only ones that can be considered for calculating damages here from the outset and at most. When calculating the amount of damages to be paid according to the principles of license analogy, the question is what reasonable contracting parties would have agreed as remuneration for the acts of use performed by the infringer in knowledge of the actual development during the infringement period. The objective value of the right of use is to be determined (OLG Köln, judgement of 31.10.2014 - 6 U 60/14 - with further references). The amount of the license fee to be paid as damages is to be assessed in accordance with § 287 (1) ZPO, taking into account all the circumstances of the individual case, according to the free conviction of the court. In doing so, the extent of the use as well as the value of the infringed exclusive right must be taken into account (OLG Köln, judgement of 31.10.2014 - 6 U 60/14) – In the present case, it is decisive that the plaintiff distributed the program version at issue here free of charge for all possible uses and thus, in substance, completely waived any monetary exploitation of its exclusive right of use. This waiver even goes so far that the plaintiff, according to clause 4 sentence 3 of the "GNU General Public License", even leaves these copies (free of charge) to persons who have received a copy of the program due to a distribution process contrary to the terms of the license. Against this background, the "objective value" of the use of the program version at issue here can only be set at zero (doubting the determination of damages according to the principles of license analogy in the case of "open source" software distributed free of charge, also Schneider, Handbuch EDV-Recht, 5th ed. [2017], X. IT-Verfahrensrecht, marginal no. 158 a.E.; cf. also OLG Köln, judgment of 31.10.2014 - 6 U 60/14 - there marginal no. 98, on the similar case of the distribution of a photograph made available free of charge under the "Creative Commons Attribution Non Commercial 2.0" license in breach of the terms of the license). The plaintiff provided the program version at issue free of charge, so that it is not apparent what economic sense a further paid licensing could have in addition (cf. on this consideration also OLG Cologne, decision of 29 June 2016 - 6 W 72/16). Since the use of the program, including public redistribution, is already possible free of charge, a further license subject to a fee would ultimately only amount to being exempted as a licensee from the - ultimately only purely formal - provisions of the "GNU General Public License" (cf. on this consideration also OLG Cologne, order of
--	--	--	--

			<p>29.06.2016 - 6 W 72/16). There are no indications that could serve as a basis for an estimate pursuant to § 287 (1) ZPO to determine the objective value of such an "exemption". There was or is also no "dual licensing" model for the program version further distributed by the defendant and which is the subject matter of the dispute here. This version of the program - and only this specific version of the program is relevant - was distributed by the plaintiff exclusively free of charge under the "GNU General Public License". Insofar as the plaintiff refers in this context to the "proprietary" program version "T Enterprise Client 3.x", which was distributed against payment, this is not identical to the program version at issue here: according to the plaintiff's own submission, the version "T Enterprise Client 3.x" contains "bug fixes and improvements", with further references). The circumstances that influence the objective value of the presumed acts of use include, in particular, a possibly ascertainable customary value of the right of use based on actually agreed licenses (OLG Köln, judgement of 31.10.2014 - 6 U 60/14 - with further references).</p>
<p>LG Bochum, Urteil v., Az. I-8 O 294/15</p>	<p>Germany</p>	<p>Bochum Regional Court</p>	<ul style="list-style-type: none"> - The plaintiff has a claim against the defendant under §§ 242 BGB, 97 UrhG for information about the period during which the defendant offered the software "T" for download and the number of students the defendant had during this period. - Pursuant to § 242 BGB, there is a duty to provide information if the legal relationship existing between the parties entails that the entitled party is in excusable uncertainty about the existence or scope of his right and the obligated party can easily provide the information necessary to eliminate the uncertainty. - The necessary special relationship exists between the parties on the basis of a claim for damages for copyright infringement (OLG München, NJW-RR 1992, 749). - The plaintiff has a claim for damages against the defendant on the merits according to the principles of license analogy pursuant to Section 97 (2) sentence 3 UrhG. - An infringement of the plaintiff's copyright is to be seen solely in the fact that the defendant made the software in dispute publicly available without license text and source code within the meaning of § 69 c no. 4 UrhG. This software is so-called open source software, the use of which is free of charge according to the H and the further development of which is permitted. However, the right of use requires compliance with the H. Accordingly, it is necessary in

			<p>particular that reference is made to the H, the license text of the H is attached and the source code is made accessible (clauses 1 and 3 of the H). It is undisputed that the defendant has not complied with these conditions of the H. Clause 4 of the H stipulates that a breach of the license automatically leads to a lapse of the license rights, so that there is unauthorized use by the defendant.</p> <ul style="list-style-type: none"> – The defendant is also responsible for the copyright infringement, because it acted at least negligently. The plaintiff only published the software under the conditions of H and, for this purpose, as shown by the screenshots submitted as Annexes K 10, made the necessary source code and the license conditions available on its website in addition to the download of the program. The defendant cannot refute this information with its insubstantial argument that the plaintiff itself had put the software on the market without source code and license conditions. – Since the plaintiff only permitted free use of its software if the provisions of the H were complied with, it is entitled to damages on the merits if this set of rules is not complied with, even if the authorized use is free of charge. If one were to follow the defendant's legal opinion, the authors of software published under the terms of the H would be virtually without rights. Why the possibility of a claim for injunctive relief on the part of the plaintiff should exclude its claim for damages - as argued by the defendant - is not apparent to the board.
LG Halle, Urteil v. 27.07.2015, Az. 4 O 133/15	Germany	Halle District Court	<ul style="list-style-type: none"> – The use of a protected computer program until the software was removed from the homepage in May 2015 constitutes an infringement of the copyright to allow the public reproduction of the computer program. The risk of repetition with regard to the copyright infringement is not eliminated either by the affidavit of the head of the defendant's computer centre submitted at the oral hearing, according to which the software in dispute was removed from the defendant's homepage in May 2015, or by the statement that it was a strategic instruction of the defendant to remove the software in dispute from the network and not to use it in the future. For one thing, the removal of the software merely represents an actual process that could be reversed at any time, and for another, the above statements lack the necessary legal protection, which is only possible through the submission of a cease-and-desist declaration with a penalty clause. (para. 13) (para. 14) – Insofar as the first infringer is granted further use of the license under the terms of the license if he ceases

			the infringement within 30 days of receipt of a corresponding notice, this is not to be understood as meaning that the licensor thereby simultaneously intends to waive a legal claim to the submission of a cease-and-desist declaration with a penalty clause by the first infringer. Even if the licensor wants to give the infringer a "second chance" to use the license, the licensor has an interest worth protecting in preventing further infringements in the long term already after the first infringement.(para.15)
LG Hannover, Urteil v. 21.07.2015, Az. 18 O 159/15	Germany	Hanover Regional Court	<ul style="list-style-type: none"> – Offering open source software on the Internet that is covered by the license terms of the GNU General Public License (GPL) constitutes copyright infringement if the software is not accompanied by the license text of the GPL and the source code is not made publicly available or made available on a data carrier.
LG Köln, Urteil v. 17.7.2014, Az. 14 O 463/13	Germany	Cologne Regional Court	<ul style="list-style-type: none"> – Anyone who further develops software licensed under the GNU General Public License and continues to license the further development under the GNU Lesser General Public License is in breach of the original license. According to section 9 GPLv2 / section 14 GPLv3, only a different license version of the same type may be selected in the event of relicensing, not a different license type. – The claim for damages resulting from such unauthorized use may have as its object the surrender of the infringer's profits, which may also include indirect sources of financing from customer, service and support services.
LG Hamburg, Urteil v. 14.06.2013, Az. 308 O 10/13	Germany	Hamburg Regional Court	<ul style="list-style-type: none"> – The defendant has made the software "netfilter/iptables" publicly accessible and distributed it in contravention of the agreement without at the same time attaching the license conditions of the GPLv2 and without making the complete corresponding source code of the software available to third parties at the same place and under the same conditions free of license fees. – The defendant has breached the contractual penalty agreement. It made the software "netfilter/iptables" publicly accessible without offering the fully corresponding source code of the software at the same place and under equivalent conditions. Making a work available to the public (Section 19a UrhG) exists if the work can be accessed interactively. The actual retrieval is irrelevant (Wandtke/Bullinger, Urheberrecht, 3rd ed. 2009, § 19a marginal no. 10). It is undisputed that the "netfilter/iptables" software in dispute was offered for download on the defendant's website. The defendant further breached its obligation to make the "netfilter/iptables" software accessible only if the complete corresponding source

			<p>code was transmitted at the same time. The software "netfilter/iptables" was offered by the defendant as part of the firmware in the object code and made available for download, although it was not listed in the source code. By offering to download the software, it made the software publicly available.</p> <ul style="list-style-type: none"> - The defendant acted negligently culpable in any case. It should have ensured that the "netfilter/iptables" software was only made publicly available in compliance with the license conditions of the GPLv2. In doing so, it could not rely on the assurance of its suppliers that the delivered goods did not infringe the rights of third parties. In any case, the defendant would have had to carry out or arrange for a suitable examination of the software offered and made available by it by means of its own examination or with the help of competent third parties, even if this entailed additional costs. - There is already an infringement of the plaintiff's rights of use under copyright law because the "netfilter/iptables" software is implemented in the firmware included by the defendant on the FANTEC 3DFHDL media player. Since the defendant did not deliver the complete source code to the corresponding object code, there is a violation of the conditions of the GPLv2, so that there is an unauthorized use. According to the terms of the GPLv2 license, anyone is permitted to use and edit the software on the basis of the license granted and each user is required to grant third parties the same rights to his or her editing. (Wandtke/Bullinger, Urheberrecht, 3rd ed. 2009, § 96c paragraphs 74 and 81 with further references). Based on the so-called copyleft principle of § 3 GPLv2, a simple right of use is only granted if the user undertakes to offer the adaptation or transformation created by him again under the conditions of the GPLv2. This is the only way to ensure the further development and improvement of the open source software offered under a GPLv2. According to § 4 GPLv2, a violation of the provisions of the GPLv2 automatically leads to a loss of all rights of use. - As the provider of the FANTEC 3DFHDL Media Player and operator of the website and domain owner on which the firmware is also offered for download, the defendant is also responsible for the infringement according to the imprint on the website www.fantec.de (Annex K 1). A new warning was necessary because the defendant's infringement of the declaration to cease and desist, which was subject to a penalty clause, created a new risk of repetition.
--	--	--	---

LG Bochum, Teilurteil v. 10.02.2011, Az. I-8 O 293/09	Germany	Bochum Regional Court	<ul style="list-style-type: none"> – An infringement of the plaintiff's rights of use under copyright law exists solely because the program "G" is implemented in the software for the program "N", as is shown by the letter of 27 March 2009 from E, the company responsible for the programming (annex K 15 a to the plaintiff's statement of 11 January 2010). In this letter, E admits that the software in dispute was inserted into the program "N" and not removed again, resulting in claims under copyright law. In view of this, it is irrelevant whether the software in dispute is functionless within the program "N"; the act of inserting this software into the program "N" is already relevant under copyright law. In any case, the non-functionality of the software distributed by the plaintiff does not follow solely from the fact that the defendant's program should be able to run without restrictions even without this software. It is undisputed that the defendant has not complied with the conditions of the Lesser General Public License, so that there is unauthorized use. – The assertion of the claim for information is also neither unfaithful nor disproportionate. The plaintiff only published the software "G" under the terms of the Lesser General Public License, which the defendant indisputably did not observe. Since the plaintiff only permits the free use of its software if the provisions of the LGPL are complied with, it is entitled to claim damages on the merits in the event of non-compliance with this set of rules according to the principles of license analogy, even if the authorized use is free of charge. If one were to follow the legal opinion of the defendant, the authors of software published under the terms of the LGPL would be practically deprived of their rights.
LG München I, Urteil v. 12.07.2007, Az. 7 O 5245/07	Germany	Munich Regional Court	<ul style="list-style-type: none"> – With the free accessibility of software on the Internet, the author does not automatically allow every third party to use his software. Mere publication also does not mean a waiver of the assertion of injunctive relief. – The author of software which he has published on the Internet under the GPL has a right to injunctive relief against users of his software who do not comply with the rules of the GPL. – The provider of a web shop platform is liable for copyright-infringing products even if he does not operate the shop himself but only provides the technical basis for it.
LG Frankfurt a.M., Urteil v. 06.09.2006,	Germany	District Court of Frankfurt am Main	<ul style="list-style-type: none"> – By distributing the data storage unit, in the firmware of which the programs mtd, initrd and msdosfs are included, without complying with the provisions of the GPL, Defendant violated the copyrights in the programs, as a result of which Plaintiff, who is entitled to exercise the copyrights, could assert a claim to

<p>Az. 2-6 O 224/06</p>			<p>cease-and-desist against Defendant (Section 97 of the German Copyright Act (UrhG))</p> <ul style="list-style-type: none"> – The GPL applies to the legal relationship between the authors and Defendant. The three software programs are undisputedly licensed only under the terms of the GPL. In the case of free software it is to be assumed that the copyright holder by putting the program under the GPL makes an offer to a determinable or definite circle of people and that this offer is accepted by users [of the software] through an act that requires consent under copyright law; in this respect, it can be assumed that the copyright holder enters into this legal relationship without receiving an actual declaration of acceptance [from the users] (Section 151 of the German Civil Code (BGB)). – The conditions of the GPL can in no case be interpreted to contain a waiver of legal positions afforded by copyright law. The GPL precisely stipulates that the freedom to use, modify and distribute the corresponding software initially afforded by way of a grant of a non-exclusive license to everyone is automatically terminated upon a violation of the GPL (cf. Dreier/Schulze, § 69a, Rz.11) – Pursuant to Sec. 4 of the GPL the rights under the GPL are terminated and revert to the author if the user violates the obligations set forth in Sec. 2 of the GPL. In particular, these obligations provide that the user has to publish a disclaimer of warranty on each copy [of the program], make reference to the GPL, accompany the program with the license text, and provide the source code of the program. – Defendant cannot invoke a claim of exhaustion of the right to distribute (Section 69 c, No. 3, Sentence 2 of the German Copyright Act (UrhG)), even though the three programs are available to the public on the Internet. The principle of exhaustion only applies to the individual physical data carrier onto which the software is copied during the downloading process. With respect to the right to copy, no exhaustion takes place; thus, Defendant is not entitled pursuant to Section 69 c, No. 3, Sentence 2 of the German Copyright Act (UrhG) to [freely] copy the software onto the individual data storage units. – With respect to the data storage units already sold by Defendant, no exhaustion of the right to distribute takes place, since those data storage units were not put into circulation by sale with the consent of the authors as the sale of the data storage units did not comply with the GPL. However, purchasers can, as described above, at any time acquire the necessary
-----------------------------	--	--	--

			rights of use [the three programs] directly from the author by recognizing the GPL.
LG Berlin, Beschluss v. 21.02.2006, Az. 16 O 134/06	Germany	Regional Court of Berlin	<ul style="list-style-type: none"> – In the event of an infringement of the GPL, the author of the software has a right to injunctive relief against the infringer. – The applicant has the exclusive distribution right to the programs "netfilter/iptables" and "mtd", which the defendant has unlawfully interfered with. It is irrelevant whether the defendant was contractually bound to the General Public License or not. In the former case, the rights of use would have reverted to the applicant in accordance with section 4 of the GPL, because the applicant distributed the programs without referring to the licensing under the GPL and without making the source code available free of charge (sections 1 and 3 of the GPL). In the latter case, the defendant would not have effectively acquired the rights of use anyway, since the "free" availability of the programs in no way constitutes a waiver of the rights of the authors and exclusively authorized users (cf. on the whole: LG München I CR 2004, 774 = MMR 2004, 693 = GRUR-RR 2004, 350).
LG München I, Urteil v. 19.05.2004, Az. 21 O 6123/04	Germany	Munich Regional Court	<ul style="list-style-type: none"> – One cannot perceive the conditions of the GPL (General Public License) as containing a waiver of copyright and related legal positions. On the contrary, the users avail themselves of the conditions of copyright law in order to secure and carry into effect their concept of the further development and dissemination of software (see Dreier/Schulze, UrhG, § 69a, Rn. 11).
TGI de Paris, 2019-06-21	France	TGI de Paris	<ul style="list-style-type: none"> – In the context of a call for tenders launched at the end of 2005 by the Agency for the Government of Electronic Administration (ADAE), subsequently attached to the General Directorate for State Modernization, with a view to the design and implementation of the portal called "Mon service Public", the company Orange SA, a telecommunications operator, was awarded lot no. 2, relating to the supply of an IT solution for identity management and interface resources, intended for service providers. – Orange proposed a solution including the interfacing of the IDMP platform with the Lasso software library, published by Entr'ouvert, a software publisher dedicated to digital identity management, and under the GNU GPL Version 2 free license. – The company Entr'ouvert considered that the provision of the free Lasso library to the DGME by Orange as part of the Mon Service Public project did not comply with Articles 1 and 2 of the free license.

			<p>Following the official report of 22 and 27 April 2011, it carried out an infringement seizure at Orange's head office, and then, by deed of 29 April 2011, summoned the company Orange before this court for copyright infringement.</p> <ul style="list-style-type: none"> – In application of the provisions of article L122-6 of the Intellectual Property Code, the acts related to the right of exploitation of the software, such as "the right to carry out and to authorize : 1/ the permanent or temporary reproduction of the software (...); 2/ the translation, adaptation, arrangement or any other modification and reproduction of the software", are expressly reserved by law to the author of the software and subject to the authorization of the latter, except for the exceptions provided for in Article L122-6-1 of the same code, when these acts of exploitation are "necessary to allow the use of the software in accordance with its intended purpose, by the person having the right to use it, including for the purpose of correcting errors. – Violation of the author's reserved rights is punishable by infringement (Article L335-2 of the Intellectual Property Code). – However, "the particular terms of use to enable the software to be used in accordance with its intended purpose, by the person entitled to use it" are arranged, according to paragraph 2 of Article L122-6-1 of the Intellectual Property Code, by contract between the parties. – In this case, the Lasso library is made available under a GNU GPLv2 free license, which consists of the licensor authorizing a user to execute, distribute and modify software, on the condition that the user respects the conditions of use set out in the corresponding license agreement. – This contract is a contract of adhesion, the clauses of which cannot be discussed and negotiated by the person who obliges himself, but which nevertheless includes, contrary to the assertions of the plaintiff, reciprocal obligations for each of the parties. – Indeed, the company Entr'ouvert grants an authorization to use the software by making it available, even if it is free of charge and any guarantee on its part is excluded, whereas the user companies must respect the terms of use as set out in the license. – In this case the GNU GPLv2 in its original version (Entrouvert Document 24) and in its unofficial French translation (Entrouvert Document 25), authorizes the licensee to "copy and distribute verbatim copies of the source code" (section 1), "modify the copy of the
--	--	--	--

			<p>Program or any part thereof creating a work based on the Program, copy and distribute such modifications" (section 2), "copy and distribute the Program or a work based on it" (section 3), "copy, modify, sublicense or distribute the Program [only] as expressly provided in the license" (section 4).</p> <ul style="list-style-type: none"> – The company Entr'ouvert accused its opponent of not having respected the terms of the license and the rights and obligations created by it, by declaring that it had only used version 0.6.3, whereas the defendants had also used version 2.2.90. The defendants also claimed to have misleadingly indicated to ADAE that the library would be an autonomous module and thus to have failed to comply with the obligations imposed by the license. – It (...) appears that the company Entr'ouvert is in fact seeking compensation for damage generated by the defendants' failure to perform obligations resulting from the license and not the violation of an obligation external to the license contract. – The solution of the dispute requires the interpretation of the free license, governing the relationship between the parties in question, in order to establish the legality or illegality of the behaviour complained of. – The relationship between Entr'ouvert and the Orange companies for the use of the license is therefore contractual in nature. – In application of the principle of non-accumulation of liability, only the basis of contractual liability can be invoked by the plaintiff, who must therefore be declared inadmissible in her action for infringement and in her ancillary claims, based exclusively on tort liability, without Entr'ouvert being able to invoke the immediate and retroactive cancellation of the license, a sanction that is specific to contractual matters.
Cour d'appel de Paris, 2021-03-19	France	Cour d'appel de Paris	<ul style="list-style-type: none"> – The company Entr'Oouvert reproaches the respondent companies for not having respected the obligations resulting from the GNU GPL v2 license agreement applicable to the Lasso software, by incorporating/encapsulating the software in a new IDMP software that they marketed alone to the State and requests that it be judged that these violations, constituting an infringement of its intellectual property rights, characterize an infringement. Thus, it bases its claims on software copyright and the tort of infringement. – The first judges considered that the company Entr'Oouvert was in fact seeking compensation for damage generated by the non-performance of

			<p>contractual obligations resulting from the license, that thus, in application of the principle of non-accumulation of responsibilities, only the basis of contractual responsibility could be invoked and that it was therefore necessary to declare inadmissible the infringement action based exclusively on tortious responsibility.</p> <ul style="list-style-type: none"> – The company Entr'Oouvert maintains that it is admissible to act on the tort basis of infringement, notwithstanding the existence of a license agreement, since it invokes the violation of this agreement. It complains that the respondent companies have violated articles 2, 3, 4 and 10 of the GNU GPL v2 license agreement and claims that these breaches constitute an infringement of its intellectual property rights over the software, the ownership and originality of which it claims. It considers that there is no reason to apply the rule of non-accumulation of liability in this matter and relies on a judgment of the Court of Justice of the European Union (CJEU) handed down on 18 December 2019, subsequent to the judgment under appeal. – Court declared the request for inadmissibility to act on the basis of contractual liability presented by the companies Orange SA and Orange Business Services to be without object.
--	--	--	---

This document is an output of the ZOOM Project (GA 101007385)



Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
<https://creativecommons.org/licenses/by-sa/4.0/>