



Annex C: Non-financial sustainability issues and solution directions

Development of a Funding Mechanism for Sustaining Open Source Software for European Public Services

April 2022

Deloitte.

Study Authors

This study was carried out for the European Commission in 2021 by [Deloitte Consulting](#) supported by [Offerman Consulting](#) for the Open Source research and analysis.

Contract: Framework Contract DI/07624 - ABC IV Lot 3

Request: Funded by ISA² 2020 Sharing and Re-Use Action (2016.31), Development of a Funding Mechanism for Sustaining Open Source Software for European Public Services, request No 477, DG DIGIT B.3

European Commission

The study was managed by Saranjit Arora (external PM & member of OSPO) and Miguel Diez-Blanco (Commission PM & OSPO Lead) from DIGIT B.3.002. It was reviewed by Gijs Hillenius (OSPO), Evangelos Tsavalopoulos (Head of Sector) and others within DIGIT B.3.

Disclaimer

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the Commission. The Commission does not guarantee the accuracy of the data included in this document. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use which may be made of the information contained therein.

More information on the European Union is available on the Internet (<http://www.europa.eu>).

© European Union, 2021-2022



The Commission's reuse policy is implemented by [Commission Decision 2011/833/EU of 12 December 2011 on the reuse of Commission documents](#).

Unless otherwise noted (e.g. in individual copyright notices), the reuse of the editorial content on this website owned by the EU is authorized under the [Creative Commons Attribution 4.0 International \(CC BY 4.0\) licence](#). This means that reuse is allowed, provided appropriate credit is given and any changes are indicated.

You may be required to clear additional rights if a specific content depicts identifiable private individuals or includes third-party works. To use or reproduce content that is not owned by the EU, you may need to seek permission directly from the respective right holders. Software or documents covered by industrial property rights, such as patents, trademarks, registered designs, logos and names, are excluded from the Commission's reuse policy and are not licensed to you.

EUROPEAN COMMISSION

Directorate-General for Informatics
Directorate DIGIT
Unit B3 — DIGIT.B3 Reusable Solutions

E-mail: DIGIT-OSPO@ec.europa.eu

Table of Contents

C	Annex: Non-financial sustainability issues and solution directions	4
C.1	The current FOSS licensing landscape	4
C.1.1	Free software and copyleft licences.....	4
C.1.2	Open-source software	5
C.1.3	Open-source vs. free software	5
C.1.4	Permissive licences.....	6
C.1.5	Reciprocity	8
C.1.6	Licence compatibility.....	9
C.1.6.1	Licence interoperability	9
C.1.6.2	The EUPL compatibility matrix.....	10
C.1.6.3	The JLA Compatibility Checker	11
C.1.7	FOSS-based business models	11
C.1.7.1	Industry pushing for permissive licences.....	13
C.1.8	Licensing choices	14
C.1.9	Non-discrimination of uses and users, and licence non-proliferation.....	16
C.1.10	A changing world: new delivery models.....	17
C.1.10.1	On the FSF's points of view	17
C.2	A new licensing paradigm: post-Open-Source	19
C.2.1	The good and the bad of open source.....	19
C.2.2	A new paradigm.....	23
C.2.3	Coherent Open Source and Coherent Licences	23
C.2.4	The post-Open-Source licence	24
C.2.5	Charged business use cases.....	24
C.2.6	Apportionment of incoming payments.....	24
C.2.7	The Operational Agreement	25
C.2.8	Licence compliance and enforcement.....	25
C.2.9	Based on contract law	26
C.2.10	Charged business use cases.....	26
C.2.11	The dual-licensing choice clause.....	27
C.2.12	A patent-free and API copyright-free zone	27
C.2.13	The post-Open-Source paradigm: current state of affairs.....	27
C.3	Another new licensing paradigm: Remuneration rights	27
C.3.1.1	Copyrights and patents	28
C.3.2	An Open world	30
C.3.3	Remuneration rights as an incentive for innovations.....	31
C.3.4	Some examples	31
C.3.5	The Open model	34
C.3.6	Implementation of the Open model.....	35

C Annex: Non-financial sustainability issues and solution directions

In this annex we present:

- an overview of the current FOSS licensing landscape (basically a copyleft/permmissive dichotomy, as explained below); and
- two new "post-FOSS" licensing paradigms:
 - a post-Open-Source paradigm to succeed the current FOSS licensing paradigm, as proposed by Bruce Perens, creator of the Open Source Definition (OSD) and co-founder of the Open Source Initiative (OSI); and
 - a system of remuneration rights to replace the current incentivisation system for creation and innovation based on copyrights and patents, as proposed by Rufus Pollock, founder of the Open Knowledge Foundation (OKF) and former fellow of the Shuttleworth Foundation.

These sections provide background and detailed information on the current licensing landscape as a cause of sustainability, as discussed in section 7.1 of the report.

Note that the material discussed in this Annex is especially relevant to the European Commission as the creator and maintainer of the European Union Public Licence (EURL).^{1,2}

C.1 The current FOSS licensing landscape

In this section we discuss the current FOSS licensing landscape through its two main licensing types, i.e. copyleft and permmissive, and how FOSS industry/science consortia and FOSS companies deploy various licensing types (including non-FOSS licences) to protect their interests, sustain their (commercial) productions, and grow their businesses.

C.1.1 Free software and copyleft licences

Free or libre software licences allow users to run, study, change and (re)distribute³ the (source) code. These so-called "four freedoms" were defined in the early 1980s in the Free Software Definition^{4,5} by Richard Stallman, initiator of the GNU Project⁶ and founder of the Free Software Foundation (FSF).⁷

Stallman is also the original author of the GNU Public License (GPL), which implements these four freedoms in copyright-based licences. Crucial in these and other so-called copyleft licences is the requirement that **the same rights must be preserved in derivative works**. Most importantly, this means that others cannot use

¹ https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf

² <https://joinup.ec.europa.eu/collection/eupl/eupl-guidelines-faq-infographics>

³ <https://opensource.org/faq#distribution>

⁴ <https://www.gnu.org/philosophy/free-sw.html>

⁵ Freedom 0, the freedom to run the program for any purpose, was later added to this list.

⁶ <https://www.gnu.org/>

⁷ <https://www.fsf.org/>

and/or modify the code and redistribute it without also making available the (modified) source code.^{8 9} This reciprocal property (building on the software means contributing back) is the key difference between copyleft licences and the permissive licences that we discuss below.

C.1.2 Open-source software

Whereas the free-software movement is highly ideologically driven¹⁰ – "software should be free"¹¹ – and takes a user-centric point of view, the open-source software movement takes a practical position and a software-centric point of view.

In the late 1990s, Eric Raymond wrote the essay (later to become a book) 'The Cathedral and the Bazaar', in which he compared two open-source development/publishing models: the 'cathedral' model, in which the source code is published with each new release (Google's Android Open Source Project (AOSP) is a current example of this), and the 'bazaar' model, in which source code is developed collaboratively online (for example the Linux kernel, and many other FOSS projects today). The idea of the latter is that opening up the source code to users, customers, co-developers and beta-testers, and actively involving them in the development process, results in better software quality (though not necessarily a better software architecture).¹²

Raymond's book, and Netscape's publication of the source code of its Communicator suite as open source, were the beginning of open-source software as an (open and collaborative) development model in business and industry. In 1998, Eric Raymond and Bruce Perens founded the Open Source Initiative (OSI).¹³ Around the same time, they published the Open Source Definition (OSD),¹⁴ based on the Debian Free Software Guidelines (DFSG), which were also written by Perens.¹⁵

The Open Source Initiative differentiated itself from the free-software movement from the start. It aimed to make open-source software (as a (collaborative) development model rather than a philosophy and a distribution model) attractive to businesses and industry in its branding, marketing and advocacy.^{16 17}

C.1.3 Open-source vs. free software

Even though the Open Source Definition is formulated in wording very different from the Free Software Definition, and in more explicit and practical terms, the two definitions are not very different from each other in their meanings. The main pillars of both are the rights for users to run, study, change and (re)distri bute the (source) code. In addition, the Open Source Definition is very explicit (practical)

⁸ <https://www.gnu.org/licenses/copyleft.html>

⁹ <https://www.gnu.org/philosophy/categories.html>

¹⁰ <https://www.gnu.org/philosophy/philosophy.html>

¹¹ <https://www.gnu.org/philosophy/shouldbefree.html>

¹² <http://oss-watch.ac.uk/resources/odm>

¹³ <https://opensource.org/>

¹⁴ <https://opensource.org/docs/osd>

¹⁵ <https://opensource.org/history>

¹⁶ <http://www.catb.org/~esr/open-source.html>

¹⁷ <https://www.gnu.org/philosophy/open-source-misses-the-point.html>

on the ways in which the code is modified, (re)distributed and made available. The Open Source Definition forbids discrimination against any person, group, type of usage or technical context of usage. It requires that these rights remain independent of additional licences, and that the software in question remains isolated from differently licensed software that may be part of the same distribution.

The consensus today is that the Free Software Definition and the Open Source Definition cover roughly the same types of licences.^{18 19} Since the ideological standpoint of the free software movement is very important to a part of the community, we use the term 'Free and Open Source Software' (FOSS) in this report, as explained in section <???.>.

OSI's Open Source Definition currently sets the standard, though. As discussed in section 2.2.2 of the report, almost all existing funding mechanisms for FOSS require the resulting source code to be published under an open-source licence "officially" recognised as such by OSI²⁰ through its License Review Process²¹ (and for good reasons).²²

C.1.4 Permissive licences

OSI's open-source marketing strategy was a success, since within a few years (corresponding to the dot-com bubble) more and more businesses and other organisations started using, producing and publishing open-source software.²³

Over time, the industry became familiar with open-source licences and organisations found ways to incorporate open source not only into their software development models but also into their go-to-market strategies. As part of this development, even though they have been around much longer, permissive open-source licences became widely used and are now supplanting the copyleft licences.²⁴ According to WhiteSource,²⁵ the share of permissive licences rose from 41 percent in 2012 to 76 percent in 2020.

¹⁸ <https://www.gnu.org/philosophy/free-open-overlap.html>. According to the FSF, the main difference now is that the category of so-called tivoised software (which cannot be updated because it is cryptographically locked to a hardware device) falls under open source but is not free. <https://www.gnu.org/proprietary/proprietary-tyrants.html>

¹⁹ <https://opensource.org/faq#free-software>

²⁰ <https://opensource.org/licenses/category>

²¹ <https://opensource.org/approval>

²² <http://oss-watch.ac.uk/resources/licdiff>

²³ <http://oss-watch.ac.uk/resources/businessofopensource#open-source-is-not-a-business-model>

²⁴ <https://www.zdnet.com/article/the-fall-of-gpl-and-the-rise-of-permissive-open-source-licenses/>

²⁵ <https://www.whitesourcesoftware.com/resources/blog/open-source-licenses-trends-and-predictions/>

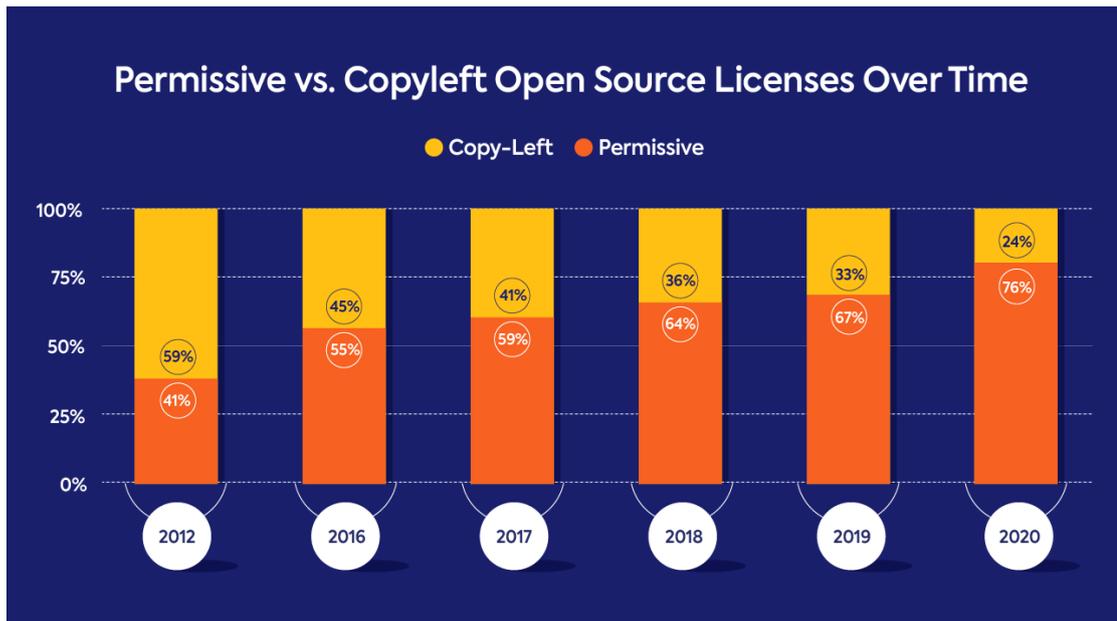


Figure 1 - Permissive vs. copyleft open-source licences over time. Source: WhiteSource²⁶

Permissive (or "BSD-like") open-source licences generally contain minimal restrictions on how the software can be used, modified and (re)distributed. They often come with little more than a copyright notice/permission and a warranty/liability disclaimer. According to WhiteSource,²⁷ the most used permissive licences in 2020 were Apache 2.0,²⁸ MIT,²⁹ and BSD.^{30 31} Unlike the copyleft licences, permissive licences are often less than a single page long.

²⁶ <https://www.whitesourcesoftware.com/resources/blog/open-source-licenses-trends-and-predictions/>

²⁷ <https://www.whitesourcesoftware.com/resources/blog/open-source-licenses-trends-and-predictions/>

²⁸ <https://www.apache.org/licenses/>

²⁹ <https://fedoraproject.org/wiki/Licensing:MIT?rd=Licensing/MIT>

³⁰ <https://opensource.org/licenses/bsd-license.php>

³¹ <https://opensource.org/licenses/BSD-3-Clause>



Figure 2 - Top 10 open-source licences in 2020. Source: WhiteSource³²

C.1.5 Reciprocity³³

There are other differences between permissive and copyleft licences, and (obviously) between licences within these two categories, but the crucial difference between the two is that copyleft licences require that **the same rights must be preserved in derivative works**, while derivatives under permissive licences **may be published with just the original copyright notice and disclaimer**. That means that code under a permissive licence can also be used in closed-source (proprietary) commercial software, as long as it contains the attribution to the original creators of the source code.^{34 35}

This lack of reciprocity – a key component of the free software ideology – is what makes permissive licences so attractive to businesses and industry:³⁶ it allows them to use open-source software in their software, without any obligation to contribute back or even publish their own modifications.

The requirement of reciprocity does *not* follow from the Free Software Definition. It is only implemented in the GPL (and other copyleft licences). As a matter of fact, most permissive licences also fall under the Free Software Definition, just as most copyleft licences also fall under the Open Source Definition (since the two definitions cover roughly the same types of licences, as discussed above).³⁷

³² <https://www.whitesourcesoftware.com/resources/blog/open-source-licenses-trends-and-predictions/>

³³ What we call 'reciprocity' here is called 'share-alike' in the context of the Creative Commons licences and other schemes that facilitate openness in a wider sense.

³⁴ <https://opensource.org/faq#copyleft>

³⁵ <https://opensource.org/faq#permissive>

³⁶ In addition to very practical reasons such as ease of use and no hassles.

³⁷ <https://opensource.org/faq#copyleft>

C.1.6 Licence compatibility

From the different types of licences we just discussed it follows that licence compatibility³⁸ is a crucial point of attention when reusing FOSS. Bringing together source code from various codebases requires the licences of each and every one of them to be compatible with the licence that you intend to use for the final software (in copyright terms: a derivative work). You cannot (legally) incorporate code in your project if that code is published under a licence with requirements that contradict the licence under which the whole will be distributed.³⁹

Generally:⁴⁰

- proprietary licences are not compatible with any other licence, i.e. proprietary software cannot be reused;
- copyleft licences are only compatible with copyleft, due to the reciprocity requirement, which is intended to be incompatible with proprietary licences, and
- as a consequence copyleft licences are also incompatible with permissive licences; and copyleft licences are often only compatible with themselves, due to the "viral property", which requires derivative works to be distributed under the same licence; and
- permissive licences are compatible with all other licences, including proprietary licences.

C.1.6.1 Licence interoperability

Depending on the exact wording and interpretation of the licences and applicable copyright law, there may be room for software under incompatible licences to be combined (meaning a less close integration than a derivative work), for example in a software distribution (a so-called aggregate)⁴¹ or as separate libraries⁴² (which are (dynamically) linked rather than mixed).⁴³

Copyleft licences that do not apply their core provisions to all types of derivative works are called weak-copyleft licences, distinguishing them from strong-copyleft licences, which include all derivative works.⁴⁴ ⁴⁵ ⁴⁶ Most often, weak-copyleft licences are more liberal in linking to software under different types of licences, but this can also concern other types of derivative works or usages.⁴⁷ ⁴⁸ ⁴⁹

³⁸ <https://www.gnu.org/licenses/gpl-faq.html#WhatIsCompatible>

³⁹ <https://www.gnu.org/licenses/gpl-faq.html#ManyDifferentLicenses>

⁴⁰ <https://joinup.ec.europa.eu/collection/eupl/licence-compatibility-permissivity-reciprocity-and-interoperability>

⁴¹ <https://www.gnu.org/licenses/gpl-faq.html#MereAggregation>

⁴² The GPL licences make an explicit exception for system libraries [1, 2]. Sources: <https://www.gnu.org/licenses/gpl-faq.html#SystemLibraryException>, <https://www.gnu.org/licenses/gpl-faq.html#WindowsRuntimeAndGPL>.

⁴³ <https://www.gnu.org/licenses/gpl-faq.html#PortProgramToGPL>

⁴⁴ <https://www.gnu.org/licenses/gpl-faq.html#GPLStaticVsDynamic>

⁴⁵ <https://www.gnu.org/licenses/gpl-faq.html#GPLPlugins>

⁴⁶ <https://www.gnu.org/licenses/gpl-faq.html#IfLibraryIsGPL>

⁴⁷ <https://www.gnu.org/licenses/gpl-faq.html#GPLModuleLicense>

⁴⁸ <http://oss-watch.ac.uk/resources/licdiff#strong-and-weak-copyleft>

⁴⁹ <https://opensource.org/faq#linking-proprietary-code>

The most notable example of a weak-copyleft licence is the GNU Lesser General Public License (LGPL), which is often used to publish software libraries under a copyleft licence while allowing (as-is) linking even to proprietary software.^{50 51 52}

The distinction between strong and weak copyleft is not applicable in the EU jurisdiction, however, since linking does not result in a derivative work under European copyright law.^{53 54 55} So instead of weak copyleft the term *interoperability* is used in the EU context (as defined in the EU Computer Programs Directive 2009/24/EC).^{56 57}

C.1.6.2 The EUPL compatibility matrix

The European Commission has published a compatibility matrix⁵⁸ showing the relationship between all OSI-approved licences and the (also OSI-approved)⁵⁹ European Union Public Licence (EUPL).^{60 61} The matrix was last updated in 2017, with the publication of the current EUPL version 1.2.⁶² It provides compatibility information on two directions:⁶³

- upstream: allowing you to incorporate/link work under another FOSS licence into a larger work that will be distributed under the EUPL; and
- downstream: allowing you to distribute under another FOSS licence a larger work that incorporates/links work that falls under the EUPL.

As an interoperable copyleft licence,⁶⁴ the EUPL is interoperable (e.g. linkable) with other FOSS licences. It is downstream compatible with most copyleft licences (a dozen explicitly listed in the compatibility list in the appendix of the licence).⁶⁵ And

⁵⁰ <https://www.gnu.org/licenses/license-list.html#LGPL>

⁵¹ <https://www.gnu.org/licenses/why-not-lqpl.html>

⁵² <https://www.gnu.org/licenses/gpl-faq.html#LGPLStaticVsDynamic>

⁵³ <https://joinup.ec.europa.eu/collection/eupl/matrix-eupl-compatible-open-source-licences#section-3>

⁵⁴ <https://joinup.ec.europa.eu/collection/eupl/news/copyleft-or-reciprocal>

⁵⁵ The fact that almost all existing FOSS licences were based on US law [De Raadt] was an important reason for the creation of the EUPL at the time. Source: <http://oss-watch.ac.uk/resources/eupl#history-of-the-eupl>. Over the last years, China has published its own series of MulanPSL licences (part of Chinese techno-nationalism), which are based on the (permissive) Apache licence and approved by OSI. Source: <https://opensource.org/licenses/MulanPSL-2.0>.

⁵⁶ <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32009L0024>

⁵⁷ <https://joinup.ec.europa.eu/collection/eupl/licence-compatibility-permissivity-reciprocity-and-interoperability>

⁵⁸ <https://joinup.ec.europa.eu/collection/eupl/matrix-eupl-compatible-open-source-licences>

⁵⁹ <https://joinup.ec.europa.eu/collection/eupl/introduction-eupl-licence>

⁶⁰ https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf

⁶¹ <https://joinup.ec.europa.eu/collection/eupl/eupl-guidelines-faq-infographics>

⁶² <https://joinup.ec.europa.eu/collection/eupl/news/understanding-eupl-v12>

⁶³ <https://joinup.ec.europa.eu/collection/eupl/licence-compatibility-permissivity-reciprocity-and-interoperability>

⁶⁴ Even though the EUPL is a copyleft licence, it was (also) designed to be used for software that can be (re)used by industry. Source: <http://oss-watch.ac.uk/resources/eupl#main-features-of-the-eupl>.

⁶⁵ https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf

it is upstream compatible with most copyleft licences, the most notable exceptions being the GNU GPL v2 and v3,^{66 67 68} and the Affero GPL and GNU AGPLv3.^{69 70}

In most cases of incompatibility, conflicts are caused by limitations on specific usages such as commercial use or distribution, by requirements on branding, advertising⁷¹ and publication/distribution, by particular/unclear wording and/or legal intricacies, or by the requirement to notify the original owner or to transfer specific rights to the original owner.⁷²

C.1.6.3 The JLA Compatibility Checker

To help you select a FOSS licence that allows you to make your software available for reuse and is compatible with the FOSS components you intend to use in your project, the European Commission has made available the Joinup Licensing Assistant (JLA). Its Compatibility Checker allows you to select and compare licences based on permissions, obligations, prohibitions, compatibility/interoperability, legal aspects, and support.^{73 74 75}

C.1.7 FOSS-based business models

All of this does not at all mean that businesses and industry are only using FOSS without publishing FOSS themselves. On the contrary: the software industry has become a massive publisher of FOSS. The companies contributing the most to GitHub are Microsoft, Google⁷⁶ and Red Hat/IBM, and there are many other big tech companies making large numbers of contributions.^{77 78 79}

In sections 2.2.1 and 3.4 of the report we have described how industry/science consortia use open-source software to **collaboratively develop non-competitive, foundational software between partners**.⁸⁰ These consortia are generally

⁶⁶ <https://www.gnu.org/licenses/quick-guide-gplv3.html>

⁶⁷ <https://www.gnu.org/licenses/rms-why-gplv3.html>

⁶⁸ <https://www.gnu.org/licenses/license-list.html#EUPL-1.2>

⁶⁹ <https://www.gnu.org/licenses/license-list.html#AGPL>

⁷⁰ The FSF provides a compatibility matrix for all the GNU licences: <https://www.gnu.org/licenses/gpl-faq.html#AllCompatibility>. Compatibility information for other licences can be found in their licences list: <https://www.gnu.org/licenses/license-list.html>. Another compatibility matrix is provided by the Open Source Automation Development Lab (OSADL): <https://www.osadl.org/fileadmin/checklists/matrix.html>.

⁷¹ <https://www.gnu.org/licenses/bsd.html>

⁷² <https://www.gnu.org/licenses/license-list.html>

⁷³ <https://joinup.ec.europa.eu/collection/eupl/solution/joinup-licensing-assistant/jla-find-and-compare-software-licenses>

⁷⁴ <https://joinup.ec.europa.eu/collection/eupl/solution/joinup-licensing-assistant/about>

⁷⁵ <https://joinup.ec.europa.eu/collection/eupl/jla-joinup-licensing-assistant-select-and-compare-open-licenses>

⁷⁶ <https://opensource.googleblog.com/2020/08/open-source-by-numbers-at-google.html>

⁷⁷ <https://www.whitesourcesoftware.com/resources/blog/the-top-10-companies-contributing-to-open-source/>

⁷⁸ <https://opensourceindex.io/>

⁷⁹ <https://www.infoworld.com/article/3253948/who-really-contributes-to-open-source.html>

⁸⁰ <http://oss-watch.ac.uk/resources/sustainableopensource>

highly professionalised, and funding apparently is not an issue as long as there is a valid business case.

And in section 3.5 we described how industry/science consortia and FOSS companies nowadays use **a whole spectrum of open-source (and non-open-source) licences in parallel**, thereby at the same time furthering and protecting their (commercial) interests. In general:

- Consortia use highly permissive licences for functionality that they deem (or declare) non-competitive (i.e. a commodity), while their members create commercial offerings of non-open-source products and services⁸¹ (a form of complementary goods) from which they derive their unique and distinctive value [NumFOCUS, Eclipse, OW2].

Sharing and reusing software components reduces time to market, facilitates business and market opportunities between members, and reduces technology, legal and market risk [OW2].

- Individual companies use FOSS licences to commoditise software that is enabling but complementary to their commercial offerings, thereby externalising the costs of development and maintenance [JetBrains].

Consortia and companies that **know how to use these mechanisms to their advantage appear to be doing well business-wise**. Typically, JetBrains, Redis Labs and Confluent (see sections B.5.3 and B.5.3.2 of Annex B) all maintain sophisticated but similar matrices of products/components and licences based on multi-tiered open-core models.

According to JetBrains, companies that can sustain themselves through the classic open-source business model – by offering services such as training and consultancy (i.e. complementary goods)⁸² – are the exception (Nextcloud, for example, is one of them).⁸³ The bulk of them will not make it without a multi-licensing model (i.e. market segmentation) or VC investment.⁸⁴ In general, less permissive or even closed-source licensing is still required to sustain an open-source business. However, according to Rufus Pollock (see section 7.1.8 of the report), most often using an open-core model results in moving everyone to the enterprise edition and starving the community edition, turning this into a freemium-like model.

A dual-licensing strategy (using different types of licences for the same piece of software)⁸⁵ is generally seen as more friendly than the open-core model (using different types of licences for different components or releases of the software).⁸⁶ Moving to a less permissive licence later on (typically either because the current licence does not allow you to sustain the project, or because a large cloud provider

⁸¹ <http://oss-watch.ac.uk/resources/businessofopensource>

⁸² Rufus Pollock (see section <C.3>) calls this the "fries and ketchup" approach: give away the fries and sell the ketchup, or vice-versa. The pitfall of this complementary-goods model is that creating and maintaining the free good (e.g. the software) may be far more expensive than the good bringing in the money (e.g. the services). According to Pollock, remuneration rights (as discussed in section C.3.3) can complement this model.

⁸³ Conference "Open Source Beyond 2020", 14–15 November 2019, Brussels

⁸⁴ VC investments are seen as a threat to the FOSS ecosystem, since they most often result in projects moving from FOSS to a closed model as part of their monetisation/commercialisation. One example is Semgrep, a tool for code analysis that moved to the source-available Commons Clause licence after the owner company r2c received a Series A round of funding in October 2020. Sources: <https://r2c.dev/blog/2020/introducing-semgrep-and-r2c/>, <https://github.com/returntocorp/semgrep-rules/pull/968>.

⁸⁵ <http://oss-watch.ac.uk/resources/duallicence2>

⁸⁶ <https://www.fsf.org/blogs/rms/selling-exceptions>

is offering your network/server software as a commercial service) will get you a lot of backlash, because you are changing the conditions for the use of technology that others have been relying on.

However, strictly speaking the last point is not true: You cannot change a software licence retroactively, since the existing codebase remains available under the old licence,⁸⁷ allowing anyone to fork and further develop the software. The new licence only applies to future versions (assuming you have a Contributor Licence Agreement (CLA)^{88 89} in place). So relicensing does not change the conditions, though it may disappoint people in their expectations, especially those who have been users or contributors and do not agree with the direction taken, or those who have been building on the software in any way.

As discussed in section 2.3 of the report, similarly to consortia and companies, the EU Framework Programmes based on openness apparently treat science and technology in itself as non-competitive. Competition, e.g. putting the ecosystem and its outcomes to profitable use, takes place at enterprise level.

However, sustainability of software produced under the Framework Programmes should be one of the requirements of the programmes. And the value of individual parts of projects is often higher than the value of the project as a whole; the EU has no consistent strategy to harvest this value. [CWI]

Also in section 2.3 we described how this project aims to best serve the (long-term) FOSS interests of European public services, and compared that to how Canonical is using the Ubuntu Foundation to fund Ubuntu Linux, which forms the basis of that company.

C.1.7.1 Industry pushing for permissive licences

Apart from the use of permissive licences for their own open-source software, industry/science consortia and businesses have a strong interest in FOSS generally being published and developed under permissive licences. After all, only this type of licence allows them to build closed-source (proprietary) solutions based on open-source software. That means that industry/science consortia and businesses have every reason to promote the use of permissive licences (at the expense of copyleft (and other) licences), and so they do.⁹⁰

In sections 2.2.1 and 3.5.2 we described how people from industry and science told us that there is little difference between copyleft and permissive licences nowadays, and that the time of less permissive licences has passed.

And in sections 3.5.2 of the report and B.5.3.2 of Annex B, we discussed some extreme cases of large cloud providers simply taking FOSS server software, turning it into a proprietary cloud service, and starting charging for it, sometimes also hijacking the name of the original project, using their market power to make it almost impossible for the owners to sustain themselves through a viable services/SaaS business.

⁸⁷ <https://www.gnu.org/licenses/gpl-faq.html#CanDeveloperThirdParty>

⁸⁸ <https://opensource.org/faq#contributor-agreements>

⁸⁹ <http://oss-watch.ac.uk/resources/cla>

⁹⁰ <https://opensource.com/article/19/2/umpires-open-source-licenses>

C.1.8 Licensing choices

Before we discuss extended (e.g. conditional), alternative and next-generation licensing models, we provide an overview of what the copyleft/permissive dichotomy described above roughly boils down to in terms of licensing choices:^{91 92}

- if you simply want to publish your source code, to be used by anyone in any way, you can select a permissive licence;
- if you want to publish and collectively develop your source code:
 - select a copyleft licence if you care about the free software ideology and/or expect reciprocity from companies building commercial solutions based on your software.^{93 94}

This does not prevent companies from using your software internally,⁹⁵ and it does not prevent them from building commercial solutions based on your software as long as they contribute back;⁹⁶

This does not prevent you either from offering different arrangements for commercial use of your software in products and services through a dual-licensing option (for code that you own, or for which you are entitled to do this through a Contributor Licence Agreement, CLA⁹⁷);^{98 99}

- or select a permissive licence if you want industry/businesses to participate in the development of your software, even though that allows them to use your software in a commercial way without ever contributing back;

If there is value in your software that can be exploited commercially, by publishing under a permissive licence you allow any company to exploit this value without ever contributing back, which will make it harder for you to build a community of volunteers around your software;

At any time another party (e.g. a business) can take over "your" project simply by pouring resources into it;

- if you develop software as a company, check whether you can incorporate open source in your business model and go-to-market strategy:
 - identify the software assets/opportunities that hold or produce your distinctive value, and the assets/opportunities that can enlarge your market and/or improve your market position;
 - think about the customer value proposition (CVP), since that is what eventually determines whether a project will be sustainable or not [CWI];
 - test use cases through proofs-of-concept [CWI];

⁹¹ <https://arstechnica.com/gadgets/2020/02/how-to-choose-an-open-source-license/>

⁹² <https://choosealicense.com/>

⁹³ <https://www.gnu.org/licenses/license-recommendations.html>

⁹⁴ <https://www.gnu.org/philosophy/pragmatic.html>

⁹⁵ <https://www.gnu.org/licenses/gpl-faq.html#NoDistributionRequirements>

⁹⁶ <https://opensource.org/faq#commercial>

⁹⁷ <https://opensource.org/faq#contributor-agreements>

⁹⁸ <https://www.gnu.org/licenses/gpl-faq.html#HeardOtherLicense>

⁹⁹ <https://www.gnu.org/licenses/gpl-faq.html#ReleaseUnderGPLAndNF>

- create a differentiated licensing regime that brings non-competitive software assets (or assets designated as such) under a FOSS licence, and keeps proprietary any (complementary) assets/opportunities that have commercial value; a shared foundational framework does not have to be sustainable on its own, since there is no market demand for the framework itself; so an initial investment will be required to get the project started and take it to the point where it is sustainable [CWI];
- note that using a reciprocal (copyleft) licence guarantees that others cannot create derivatives that compete with your commercial offerings,^{100 101} which makes this type of licence the preferred choice if you are doing (financing) most of the development of the open-source code yourself;
- for a consortium, a permissive licence for non-competitive software (functionality) is the preferred choice, as it guarantees the availability of a shared/common foundation and provides each participant (and others) with the opportunity to build their own (proprietary) commercial solutions on top of this foundation; commitment of the partners of a consortium should be to the project (i.e. contributive), not to the products (extractive) [CWI];
- since you can not build a business on volunteers working in the evening hours (for example, to fix bugs), there should be a budget for maintenance [CWI]; a good way to contribute would be for employers to make available hours that their employees can work on FOSS that the company benefits from; that is also the best way to get developers involved in critical FOSS projects [CWI, De Raadt];¹⁰²
- if you as a government organisation want to publish and collectively develop (govtech) software, you can select a reciprocal (copyleft) licence to prevent companies from offering derivatives as (proprietary) commercial solutions¹⁰³ and encourage other government organisations to participate in the collaborative development efforts.¹⁰⁴

However, according to Theo de Raadt, founder and lead developer of the OpenBSD and OpenSSH projects, in practice it does not make a difference whether a project uses a reciprocal or a permissive licence (hence the type of licence will not affect its sustainability): Cisco will not spend a penny on FOSS either way; its membership of the Linux Foundation is strictly for the appearance, and at best only a tiny fraction of their membership fees will actually reach only the top of the FOSS developers.

Furthermore, if Cisco incorporates a piece of GPL-licensed code in the software stack of its switches/routers and makes its own sources available, that will not help you fix your Cisco device: you cannot reproduce their build environment, you cannot replace parts of the whole, and you cannot digitally resign the binary images so they will boot.¹⁰⁵

¹⁰⁰ <https://www.gnu.org/licenses/gpl-faq.html#WhyUseGPL>

¹⁰¹ <https://www.gnu.org/licenses/gpl-faq.html#GPLInProprietarySystem>

¹⁰² This was, for example, part of President Barack Obama's 2012 strategy to connect citizen developers to govtech FOSS and open data. [1, 2]

¹⁰³ <https://www.gnu.org/licenses/gpl-faq.html#WhyUseGPL>

¹⁰⁴ <https://www.gnu.org/licenses/gpl-faq.html#SwitchToLGPL>

¹⁰⁵ This is what Richard Stallman calls tivoisation: software that cannot be updated because it is cryptographically locked to a hardware device; it falls under open source but is not free. Source: <https://www.gnu.org/proprietary/proprietary-tyrants.html>.

So De Raadt refers to FOSS as 'software given away' ("*on the money side, volunteer open source builds its own race to the bottom*") and prides himself in producing highly-secure gold standard pieces of software [compare reference implementations], which land in commercial products and make the world a better place. In this case, however, the licensing type is important: even though the fact that OpenBSD and OpenSSH are (permissively) licensed under BSD is mostly historical, OpenSSH would not be in every network device if the software would have been reciprocally licensed ("*in our specific little corner, non-permissive is a non-starter*").¹⁰⁶

C.1.9 Non-discrimination of uses and users, and licence non-proliferation

Both the FSF and the OSI oppose developers setting additional conditional requirements when publishing their software under a FOSS licence. <TODO: source FSF>^{107 108} This so-called discrimination against specific uses and/or users of the software goes against both the Free Software Definition and the Open Source Definition. Many of the licences the FSF categorises as *nonfree* (or *not free*) forbid specific uses and/or users (as discussed in section C.1.6.2).¹⁰⁹

Note, however, that the GPLv3 in section 7 offers an exception mechanism that allows creators/contributors to set additional terms with regard to warranties/liabilities, attribution and trademarks.¹¹⁰

The two organisations also speak out strongly against licence proliferation, i.e. the creation of new FOSS licences, often to satisfy very specific needs.^{111 112}

These standpoints are completely understandable from their point of view: it goes against their foundational definitions of free software and open-source software, and it is burdensome in terms of testing licences against these definitions, evaluating the legal implications and maintaining compatibility matrices.

Furthermore, a modular approach in which licences can be extended with – or even composed of – usage/user-specific clauses (conditions) – compare the Creative Commons licences¹¹³ – would create a patchwork of incompatible licences. The FSF (e.g. Stallman) calls it a "disastrous path" that would "wreck the free software community" and "push users towards nonfree software".¹¹⁴

To reduce licence proliferation, in 2013 GitHub launched the website 'Choose an open source licence',¹¹⁵ which now presents the (permissive) MIT licence and the

¹⁰⁶ Email exchange with Theo de Raadt in June-July 2021

¹⁰⁷ <https://opensource.org/faq#restrict>

¹⁰⁸ <https://opensource.org/faq#evil>

¹⁰⁹ <https://www.gnu.org/licenses/license-list.html#NonFreeSoftwareLicenses>

¹¹⁰ <https://www.gnu.org/licenses/gpl-3.0.html>

¹¹¹ <https://www.gnu.org/licenses/gpl-faq.html#ModifyGPL>

¹¹² <https://opensource.org/proliferation>

¹¹³ <https://creativecommons.org/licenses/>

¹¹⁴ <https://www.gnu.org/philosophy/programs-must-not-limit-freedom-to-run.html>

¹¹⁵ <https://www.infoworld.com/article/2611422/github-finally-takes-open-source-licenses-seriously.html>

(copyleft) GPLv3 as the main options.^{116 117} And in section C.2.3 we will discuss the Coherent Licence Set proposed by Bruce Perens as part of his post-Open-Source paradigm. This consolidated set of mutually compatible licences consists of Apache 2.0, LGPL version 3 and Affero GPLv3.

C.1.10 A changing world: new delivery models

Even though we understand the practical implications of discrimination against specific uses and/or users of the software in general, we think that some of the arguments brought forward by the FSF are unfair, unbalanced or outdated.

C.1.10.1 On the FSF's points of view

The exclusions given by the FSF as examples of discrimination concern military use, torture, fraud, religious beliefs, commercial use, animal testing, ESC experiments, illegal copying, taste in music and food, and the use of information.^{118 119}
¹²⁰ Except for commercial use, these examples have nothing to do with new delivery models, developments in the IT market, or changing business practices in the industry, which we think should be taken into consideration by both developers and those responsible for the licensing landscape.

As a matter of fact, the publication of the GNU Affero General Public License (GNU AGPLv3) in 2007,¹²¹ based on but distinct from the GPLv3,^{122 123} makes it clear that strict adherence to the principle of non-discrimination against specific uses and/or users of the software is not tenable, even without taking into account developers'

¹¹⁶ <https://choosealicense.com/>

¹¹⁷ Much earlier, in 2006, the OSI published the (now outdated) Licence Proliferation Report as part of its Licence Proliferation Project. Source: <https://opensource.org/proliferation>. It suggested using one of nine 'Licences that are popular and widely used or with strong communities'. Source: <https://opensource.org/proliferation-report>. The categorisation of licences developed at the time is still being used: <https://opensource.org/licenses/category>.

¹¹⁸ <https://www.gnu.org/philosophy/programs-must-not-limit-freedom-to-run.html>

¹¹⁹ <https://www.gnu.org/licenses/gpl-faq.html#NoMilitary>

¹²⁰ The Ethical Source movement rejects the use of FOSS for mass surveillance, anti-immigrant violence, protester suppression, racist policing, the deployment of cruel and inhumane weapons, and other human rights abuses. It builds tools to enforce fair, ethical, and community-minded terms for those who benefit from or are affected by their work. The movement has defined seven principles, which point to reciprocal licence types extended with conditions for ethical applications.

Some examples of existing (nonfree) licences that include ethical requirements on users and/or usage are the Anti-Capitalist Software License, the Hacktivismo Enhanced-Source Software License Agreement (HESSLA) and the original JSON License.

Apart from definition problems and the question of legitimacy, Bruce Perens (and others; see section [C.2](#)) does not think these types of conditions could ever work in court.

Rufus Pollock (see section C.3), who advocates openness in a far wider sense, thinks that a myriad of restrictions would make the system unwieldy, and that the accumulation of specific conditions would be highly detrimental to creativity. Source: his 2018 book '[The Open Revolution](#)'.

¹²¹ <https://www.fsf.org/news/agplv3-pr>

¹²² <https://www.gnu.org/licenses/license-list.html#GNUGPLv3>

¹²³ <https://www.gnu.org/licenses/license-list.html#AGPLv3.0>

interests. The AGPLv3¹²⁴ is a copy of the GPLv3, extended with a clause that requires reciprocity if you run a modified program on a server and let other users communicate with it there (i.e. making only its functionality available, without the technical need to distribute the software itself in any way).^{125 126} The FSF recommends using this licence for server software whose functionality can be offered online, for example through a web interface, using Software-as-a-Service (SaaS) as the delivery model.¹²⁷

At the same time, the FSF admits that there is no obvious licensing solution to prevent companies from offering (online) business services where they never offer the software itself or its functionality as such, but instead process data on their systems on behalf of their customers. The FSF calls this Service-as-a-Software-Substitute (SaaS), and advises against using it, because users have no control over what happens to their data. Even if the source code were made available for users to check the correct processing of their data, they can never know whether their data was copied during the process.¹²⁸

The FSF even goes so far as to say that for this reason software that is meant specifically and only for SaaS should not be written at all.¹²⁹ Note, however, that processing open data this way would be a valid use case.

So a software philosophy that starts with a Free Software Definition that includes a clause on non-discrimination against specific uses and/or users of the software results in a series of licences that require reciprocity (which is perfectly reasonable and falls within the Definition). But from the source code the philosophy then expands first to the functionality and then to the business services that can be build on it, and ends with a class of forbidden software.¹³⁰

And more new undesired use cases will come up as technology progresses and new delivery models and derivatives are developed.¹³¹ For example, when Microsoft in June 2021 announced its Copilot software development tool¹³² (based on mining source code hosted on GitHub), one of the issues that FOSS developers brought up was whether GPL-licensed software could be used to feed/train a proprietary tool like this. Note that in this case, the source code is used not to create new software directly, but as input data to create a smart tool that provides extensive code-completion functionality to software developers.¹³³

For Stallman, free software is all about taking control over end users (through the software) away from the developers/owners, and moving it to the users. He argues that the power that the developers/owners of proprietary/nonfree software have over end users corrupts them, and even goes as far as to call most proprietary software these days malware.¹³⁴ The FSF publishes an extensive list of examples,

124 <https://www.gnu.org/licenses/agpl-3.0.html>

125 <https://www.gnu.org/licenses/license-list.html#AGPL>

126 <https://www.gnu.org/licenses/gpl-faq.html#UnreleasedModsAGPL>

127 <https://www.gnu.org/licenses/why-affero-gpl.html>

128 <https://www.gnu.org/philosophy/who-does-that-server-really-serve.html>

129 <https://www.gnu.org/licenses/why-affero-gpl.html>

130 <https://www.gnu.org/licenses/why-affero-gpl.html>

131 <https://denisnazarov.com/what-comes-after-open-source>

132 <https://copilot.github.com/>

133 <https://tweakers.net/nieuws/183800/github-brengt-ai-programmer-uit-die-helpt-bij-het-schrijven-van-code.html>

134 <https://www.gnu.org/philosophy/free-software-even-more-important.html>

mostly of cases of mass surveillance, censorship (as also discussed in section <4.5.7>), malicious business practices, and dark patterns.^{135 136}

However, when the FSF writes that "free software means software controlled by its users, rather than the reverse" and that "the proper job of software licences is to establish and protect users' freedom",¹³⁷ we think that this is too much about software and its users exclusively, and that developers (the original creators of the source code and owners of the copyright) and their interests (including sustainability) fall victim to this one-sided view.

C.2 A new licensing paradigm: post-Open-Source

Bruce Perens, creator of the Open Source Definition (OSD) and co-founder of the Open Source Initiative (OSI), as described in section <C.1.2>, has been working on a new (non-FOSS) licensing paradigm that could succeed the current FOSS licensing paradigm.

The main parts of this post-Open-Source paradigm are:

- the consolidation of all OSI-recognised FOSS licences¹³⁸ into three existing, mutually compatible licences: the so-called Coherent Licences (Apache 2.0, LGPL version 3, Affero GPLv3);
- the creation of a new post-Open-Source licence that:
 - combines a dual-licensing scheme into a single licence;
 - implements a scheme (using payment processors) for collecting licence fees from business users, and distributing incoming payments among the developers; and
 - allows existing FOSS projects to partake in this new paradigm (and its developers to get paid) by dual-licensing their software with the post-Open-Source licence; and
- the creation of Operational Agreements between the developers of individual projects, which:
 - specify how payments will be apportioned among the developers, and
 - allows any developer to sue a business user who does not comply with the post-Open-Source licence.

We summarise and paraphrase Perens' proposal and sentiments below.^{139 140}

C.2.1 The good and the bad of open source

Looking at what has been achieved, Perens celebrates the success of open-source software: There is a lot of good open-source software, and it is everywhere. A lot

¹³⁵ <https://www.darkpatterns.org/>

¹³⁶ <https://www.gnu.org/proprietary/>

¹³⁷ <https://www.gnu.org/philosophy/programs-must-not-limit-freedom-to-run.html>

¹³⁸ <https://opensource.org/licenses/category>

¹³⁹ Video presentation 'What Comes After Open Source', by Bruce Perens, for DebConf20, 24 August 2020 [1, 2]. Source: <https://www.youtube.com/watch?v=vTsc1m78BUk>.

¹⁴⁰ Interview and mail exchanges with Bruce Perens in February–June 2021

of people get paid to produce open-source software. And businesses are increasingly contributing directly to open-source software, rather than indirectly through consortia.

There are, however, also many ways in which open source is broken, and the community has been talking about these issues for decades without solving them.

Open source and businesses have not been a good match, because of the power differential between the two (compare the lack of leverage we discuss in section 7.1.3.2 of the report).

Open source works very well for business (open source follows business, which has the money), but fails to achieve any of its other goals.

Perens here refers to the ideology of the free software movement, which is about freedom and good for people.¹⁴¹ But all of the good has never reached the general population and today they need it more than ever. They do use open-source software but most of the time they are not aware of it. And the software is used to exploit them, e.g. by businesses who collect information about users and even manipulate and lie to them.

Perens emphasises that this is not the end users' fault; it is the community's fault for not serving their needs. "We should give people software that serves their needs and respects their civil rights."

He does not use the word leverage (as we do in section 7.1 of the report), but reminds us that it is FOSS developers who write the software that keeps the whole world running. Yet in this power constellation they are supplicants who have to beg to sustain their projects,¹⁴² e.g.:

- please give me a job at your exploitative company, so I can work on open-source software (where developers have no power), or
- please sponsor me on Patreon (which generally does not work, as discussed in section 3.1.1 of the report).

The example he gives is OpenSSL, which was maintained by a single person. It had to come to the Heartbleed security bug before the Linux Foundation stepped in and set up its Core Infrastructure Initiative (CII; see section 4.4). The Heartbleed bug was also the reason for the European Parliament to instigate the EU-FOSSA initiative (see section A.1.2 of Annex A). In section 4.4 of the report we discuss how CII, now part of the Open Source Security Foundation (OpenSSF), seems to have lost its momentum.¹⁴³ [De Raadt]

¹⁴¹ For example, in August 2021, the Fedora Linux project (<https://docs.fedoraproject.org/en-US/project/>) was recognised as a public digital good by the Digital Public Goods Alliance (DPGA), a multi-stakeholder initiative aiming to accelerate the Sustainable Development Goals (<https://sdgs.un.org/goals>) in low- and middle-income countries through digital public goods such as FOSS. Source: <https://fedoramagazine.org/fedora-linux-earns-recognition-from-the-digital-public-goods-alliance-as-a-dpg/>.

¹⁴² Rufus Pollock (see section C.3) speaks of "*crumbs from the table of robber barons such as Microsoft, Google and Facebook*". Source: interview in February 2021.

¹⁴³ At least until now. In October 2021, Google announced it will contribute 1 million USD for funding to the Linux Foundation's Secure Open Source (SOS) programme. Source: <https://security.googleblog.com/2021/10/introducing-secure-open-source-pilot.html>. A few weeks later, the Linux Foundation announced that it had raised 10 million USD from industry to support OpenSSF. Source: <https://openssf.org/press-release/2021/10/13/open-source-security-foundation-raises-10-million-in-new-commitments-to-secure-software-supply-chains/>. Note that this initiative has been instigated by the US White House Executive Order on Improving the Nation's Cybersecurity.

According to Perens, a lot of value – and wealth – has been created, but mostly has not ended up with the developers. Freedom and wealth for the general population is a good mission, and it should not be the community's mission to give Jeff Bezos (the founder of Amazon) more freedom, wealth and power, while its members themselves do not enjoy the same.¹⁴⁴

The FOSS community is represented by powerful companies, and they set the policies for the FOSS world. The Board of Directors of the Linux Foundation, for example, consists of big tech companies,¹⁴⁵ which is why according to Perens they represent the interests of the industry:¹⁴⁶

- One of them, the Qualcomm representative, fought OSI at the ETSI standardisation institute to avoid royalty-free standards (which would allow open source);
- Another company on the Board was a major infringer of the Linux licensing terms in its main product, and may even still be;
- The Linux Foundation removed the community member seats from the Board;¹⁴⁷
- The Linux Foundation educates people about licence compliance, while at the same time deterring and defunding enforcement, so that companies do not take it seriously.¹⁴⁸

Something similar applies to the Open Innovation Network (OIN),¹⁴⁹ a patent holding company (PHC) that pools and licenses Linux-related patents. It was founded and is controlled by some of the largest patent holders in the industry (e.g. Google, IBM (now including Red Hat), NEC, Philips, Sony and Toyota). According to Perens it exists as much to protect the existing software patent system (an important revenue generator) from FOSS-driven reform as it does to protect Linux from software patents.¹⁵⁰

The community itself cannot afford to build its own (defensive) patent portfolio. Developers/groups cannot even afford to file patents on their own inventions.

OIN is not viable as a defence against patents trolls (NPEs).¹⁵¹

¹⁴⁴ <https://ottertune.com/blog/overpaying-jeff-bezos-for-aws-databases> (sarcastic)

¹⁴⁵ <https://www.linuxfoundation.org/about/board/>

¹⁴⁶ According to Theo de Raadt, industry consortia are a bigger threat to the FOSS ecosystem than individual tech companies, because the consortia drain the limited financial resources available to the community to themselves.

¹⁴⁷ This in 2016 in coincidence with Karen Sandler, Executive Director of the Software Freedom Conservancy (SFC), running to be elected by the community. The SFC (<https://sfconservancy.org/>) is a non-profit organisation providing infrastructural and legal services to FOSS projects, including copyleft compliance and enforcement actions (<https://sfconservancy.org/copyleft-compliance/>). Sources: 1, 2.

¹⁴⁸ The Linux Foundation stopped funding the SFC in 2015, around the same time when the SFC participated in Christoph Hellwig's GPL enforcement case against VMware (a member of the Linux Foundation). Sources: 1, 2, 3, 4.

¹⁴⁹ <https://openinventionnetwork.com/>

¹⁵⁰ Note that the proposed Directive on the patentability of computer-implemented inventions, which would make possible software patents under European law, was rejected by the European Parliament in July 2005. That means that computer programs as such are not patentable, only those that implement a technical invention.

¹⁵¹ Although someone involved said that OIN actually puts considerable effort in fighting NPEs.

Developers/groups cannot afford to defend themselves in patent lawsuits. They, again, are supplicants who have to beg from companies and law firms.

The patent quality improvements that the FOSS community won in the early 2000s have been reversed.

The problem of copyrights on application programming interfaces (APIs, i.e. interfaces between software components) was averted¹⁵² in April 2021 when the US Supreme Court ruled that Google's use of Oracle's Java APIs in its Android operating system was within the bounds of fair use.^{153 154 155}

Licence compliance was never designed into the FOSS paradigm;¹⁵⁶ it is expensive to enforce and the legal system dissuades those who would try.

The tools from WhiteSource do not search for snippets of FOSS code in other source code. Their marketing stresses that scanning is a bad idea.

The tools from Black Duck¹⁵⁷ (part of Synopsys) and Revenera¹⁵⁸ (part of Flexera) are very expensive and do not handle scanning large code bases well. According to Perens, it can take an entire year to set up and fine-tune these tools.

You can only enforce a licence that makes significant requirements if you are able to sue infringers, but FOSS developers cannot afford that.

There are too many FOSS licences:

- Most new licences do not add enough value to justify the additional combinatorial cost (see licence compatibility, as discussed in section C.1.6).
- Many approved or candidate licences are not in the interest of the user or the wider developer community; most are designed to favour one company.

Perens blames OSI for accepting troubling licences, and too many of them. At the same time he believes that it is a bad idea for OSI to replace the Open Source Definition (OSD) with something else.

OSI itself has set up a committee to research a successor to the OSD and a process to get there. And there are many ideas for new licence types available online.¹⁵⁹ There are also plenty of warnings and indications that the community will move on

152 <https://www.infoworld.com/article/3614693/what-the-heck-does-the-google-vs-oracle-decision-mean.html>

153 <https://www.copyright.gov/fair-use/more-info.html>

154 <https://www.eff.org/cases/oracle-v-google>

155 https://www.supremecourt.gov/opinions/20pdf/18-956_d18f.pdf

156 Enforcing compliance to a FOSS licence seldom means taking the offender to court to claim damages; most often it is enough for the community to convince (and help) the offender to do the right thing and publish the modified source code (a practice also formalised in the GNU GPLv3). According to Armijn Hemel, owner of Tjaldur Software Governance Solutions, companies are already putting a lot of effort into FOSS compliance, often as part of tracking the origins and authenticity of the software for security reasons. Source: interviews and exchanges Armijn Hemel, February/September 2021.

157 <https://www.blackducksoftware.com/>

158 <https://www.revenera.com/>

159 <https://opensource.com/article/20/9/open-source-definition>

to new, non-FOSS and post-FOSS licensing models, with or without OSI, under the open source name or under a new name.^{160 161}

C.2.2 A new paradigm

Perens believes that these problems cannot be solved by fixing FOSS. The concepts behind free software are more than 35 years old, and those of open-source software date back more than 20 years. So a new paradigm is needed. This also means that this new paradigm will not replace open source; instead, the two will coexist.

Perens' proposal is based on two principles:

- to preserve open source and its core values, and
- to offer open-source developers all the advantages of this post-Open-Source paradigm, while at the same time supporting them in continuing to allow their software to be used under open-source licences.

The new post-Open-Source licence he proposes is not a FOSS licence, since it does not match the Free Software Definition and the Open Source Definition (as discussed in sections [C.1.1](#) and [C.1.2](#) respectively).

C.2.3 Coherent Open Source and Coherent Licences

Perens introduces a new promotor of open source, which is called Coherent Open Source and resides on the domain LicenseUse.org.¹⁶² This new organisation will be the successor to OSI, but will preserve the current Open Source Definition. The list of open-source licences will be reduced to only three licences, however, each with distinct characteristics.

These Coherent Licences:

- are all approved by the FSF and OSI;
- are all compatible with each other;
- all have a clause on software patenting;
- each fits a different business purpose; and
- two of them will require more from companies than (permissive) open-source licences do today.

The Coherent Licence Set Perens proposes consists of:

- Apache 2.0: the most used permissive licence;
- LGPL version 3 (GNU Lesser General Public License): a copyleft licence aimed at software libraries, allowing linking to proprietary components; and
- Affero GPLv3: a copyleft licence that also requires reciprocity for use over a network (as discussed in section [C.1.10](#)).

¹⁶⁰ <https://opensource.org/node/1099>

¹⁶¹ <https://techcrunch.com/2019/05/30/lack-of-leadership-in-open-source-results-in-source-available-licenses/>

¹⁶² <https://licenseuse.org/>

Since the LGPL version 3 and Affero GPLv3 share a substantial part of their texts (both licences are derived from the GPLv3),^{163 164} Perens argues that actually only 2.5 licences will remain and will need to be maintained.

The idea is that Coherent Open Source will evangelise FOSS developers to use these licences, and migrate to them where possible. In the long term this will reduce complexity, increase compatibility, facilitate reciprocity and make compliance easier.¹⁶⁵

C.2.4 The post-Open-Source licence

Perens then sets the contours of a new licensing paradigm that aims to preserve the good parts of FOSS while attempting to fix some of the things FOSS failed in. He wants to keep:

- the freedom to use, modify and redistribute the code; and
- the non-discrimination of uses and users.

There will be *one* new licence and *one* new Operating Agreement, and the minimum organisation required to produce and evangelise them.

The licence will differentiate free (as in freedom) from zero cost (as in gratis), and bring together a dual-licensing combo (consisting of a reciprocal licence and a commercial licence) into a single licence.

C.2.5 Charged business use cases

To fix the power mismatch between the FOSS community and business, some business use cases will be charged. For example, companies that do not want to share their source code can pay for their use. At the same time compliance should become much easier:

- Companies will be required to send one payment processor one cheque once a year for all of the post-Open-Source software they use in their business. That will allow them to combine this software with proprietary software.
- It needs to be trivially simple to calculate the cost for charged business use cases. Simple tools should be provided to detect the packages a company uses, and this should be submitted, together with basic information about the total revenue of these charged business uses and the size of the company.

C.2.6 Apportionment of incoming payments

Developers should not be supplicants who have to beg to develop FOSS and to sustain their projects and themselves.

They should have the means to prosecute infringers and represent themselves, rather than be represented by industry stand-ins (who have very different interests).

¹⁶³ <https://www.gnu.org/licenses/license-list.html#LGPLv3>

¹⁶⁴ <https://www.gnu.org/licenses/license-list.html#AGPLv3.0>

¹⁶⁵ Bruce Perens even goes so far as to admit that favouring the permissive licence type for open source was a mistake.

A technical mechanism should be developed for the apportionment, i.e. distributing the incoming payments among the developers. That starts with the proper attribution of the original creators/contributors of the source code using git (distributed version control).

The apportionment itself should be done by the payment processor, not by the business user. This mechanism will never be perfect but should be good enough.

There will have to be rules against developers abusing the system (which may require the involvement of the community).

The payment processors themselves will get paid for their work through a small fee taken from the transactions they process.¹⁶⁶

C.2.7 The Operational Agreement

There will be one Operational Agreement between the developers of each individual project (compare a Contributor Licence Agreement, CLA¹⁶⁷). Enforcement of the Agreement will be done by the developers together.

The Operational Agreement is of no concern to the end users of the software.

The Operational Agreement sets out how payments will be apportioned among the developers of a project. This requires that developers are diligent in preserving the attribution of work that they incorporate, so other developers will be paid. Technical standards will have to be developed for carrying out this scheme, e.g. tracking who wrote what, and how much of their code survives into a specific version.

Scanning for code contributions and documentation will not be a problem. Quantifying the work of user interface designers, architects and project managers may be a problem. One solution would be to pay some roles from programmer revenue, to be decided by the individual projects.

C.2.8 Licence compliance and enforcement

Compliance has been built into the post-Open-Source paradigm, without the use of a central committee to enforce it: If you violate the Operating Agreement (a project-specific (i.e. decentralised) contract), other parties to the Agreement can get together to sue you.

Compliance for post-Open-Source software means:

- publishing all modified source code, e.g. via a git server or on the Coherent Open Source hosting platform;
- conveying the licence and statement of origin if you redistribute the source code; and
- combining with proprietary software and keeping the result private is a use case that should be paid for.

Any developer who has signed the Operational Agreement can sue a business user on behalf of all developers. Following a demand, a business user has 90 days to pay without additional charges.

¹⁶⁶ Rufus Pollock (see section [C.3](#)) notes that governments have been collecting and (re)distributing money for thousands of years, so setting up new organisations and mechanisms should be avoided if possible.

¹⁶⁷ <http://oss-watch.ac.uk/resources/cla>

C.2.9 Based on contract law

A fundamental difference between the implementations of FOSS and post-Open-Source is that the post-Open-Source licence will be based on contract law instead of copyright law.

Users without a paid business use case consent passively (just as they do now), and users with charged business use cases must consent actively online (via a payment processor).

C.2.10 Charged business use cases

These are the charged business use cases under the post-Open-Source licence:

- inclusion in a product that generates a revenue payment for its sale or direct use, e.g. commercial software, embedded devices and SaaS (inclusion under the post-Open-Source licence is defined very widely: it can also be the operating system underneath the software providing a service, for instance);
- support vended by parties that are not substantially the copyright owners of the supported work (this is possible only under contract law: anyone not joining the contract is strictly speaking not even allowed to read the code); developers should be notified of security information on a timely basis (closing a loophole where Red Hat forbids RHEL support customers to redistribute support information); and
- combination with proprietary software for internal use (which is gratis under current FOSS licences).

Companies will have to pay an annual fee based on their size, but they can subtract what they pay for other use cases.

For charged business use cases, the post-Open-Source licence will require a small portion (e.g. a percentage), depending on the total revenue¹⁶⁸ of these business uses and the size of the company. The company will have to fill out a statement and send a payment to one of the payment processors, which will handle the apportionment among the developers.

For the implementation, Perens refers to the film industry, which he calls experts at processes like this. He thinks that a simple system like this will even be less expensive for many companies than the current cost of compliance.

Companies that have paid their fees this way are free to include the software in their products and services, or in their internal toolsets.

Companies can be subjected to an audit (basically running a scan of their software in their presence), but at the same time there will be limits to protect these companies from abusive behaviour.

The vast majority of individual users, and most business users, however, will never even have to read the licence. Only if you modify or combine the software, sell it, offer it as SaaS, sell devices, or offer support, would you have to set up this compliance process.

¹⁶⁸ This portion should be taken from the revenue collected by the company from the end users of these charged business uses. A fee based on the size of the company will not work, since large companies can circumvent this using a separate, smaller entity holding the IP. And the same is true for a fee based on profits, which can easily be manipulated through creative bookkeeping.

C.2.11 The dual-licensing choice clause

The dual-licensing choice clause states that if software under the post-Open-Source licence is combined with software that is licensed under a post-Open-Source licence *and* any other licence, the licence choice for the combination must be the post-Open-Source licence. Here, combining with the post-Open-Source licence is used to force users to the post-Open-Source licensing conditions if a FOSS developer also publishes their software under a post-Open-Source licence (i.e. dual-licensing), which will get them paid for their work.

C.2.12 A patent-free and API copyright-free zone

The post-Open-Source licence creates a patent-free and API-copyright-free zone. If you bring a patent or API copyright lawsuit regarding any software under the post-Open-Source licence, all of your post-Open-Source licences terminate, regardless of what you have paid.

C.2.13 The post-Open-Source paradigm: current state of affairs

At this stage Perens has only sketched the concepts of this new post-Open-Source paradigm (as just described). The details, such as the fees and precise rules, have not been determined yet. The licence and Operational Agreement will have to be created by a lawyer specialising in FOSS licences.¹⁶⁹

Perens concludes by saying that the developers have the power, because of their numbers and their productivity [and the initial ownership of their creations under copyright law]. They need to harness that power for the good of the general population, and for the good of themselves.

C.3 Another new licensing paradigm: Remuneration rights

Rufus Pollock, founder of the Open Knowledge Foundation (OKF) and former fellow of the Shuttleworth Foundation (among many other things), is a strong advocate for radical openness, not unlike what the Shuttleworth Foundation is investigating (as discussed in section A.3.3 of Annex A): "a global open knowledge society with unhindered access to essential information and limitless opportunities for innovation and replication".

In his 2018 book 'The Open Revolution',¹⁷⁰ Pollock makes a plea for an "Open world" in which the incentivisation system for creation and innovation currently based on copyrights and patents is replaced by a system based on remuneration rights.

¹⁶⁹ Bruce Perens has registered the trademarks and written some draft licences. He said he would love to be able to hire Heather Meeker (<https://heathermeeker.com/>) to help with the (legal) implementation, but is currently working alone on these new ideas: it is hard to get funding for this, since it goes against the interests of the industry. For the same reason, Perens expects fierce rejections of his ideas from industry.

¹⁷⁰ <https://openrevolution.net/>

We summarise/paraphrase Pollock's ideas below.^{171 172}

C.3.1.1 Copyrights and patents

Copyrights and patents are temporary market monopolies on information and knowledge which are provided by society to incentivise creation and innovation.

Pollock addresses several main issues with the current systems of these two types of intellectual property rights (IPR):

- **The current terms of the protection are too long, especially for copyrights.**

Where patents generally are relatively short, typically lasting 20 years including extensions, copyrights last till between 50 and 100 years after the original creator's death. For the US and the EU this term currently is 70 years.

Within EU jurisdiction, for sound recordings this term used to be 50 years after the performance/publication, but 'Directive 2011/77/EU on the term of protection of copyright and certain related rights'¹⁷³ extended/harmonised this to 70 years beyond the life of the creator. Moreover, this extension also applied to existing recordings, extending the monopoly for back catalogues.

Several similar (retroactive) extensions were adopted in the US over the past decades, basically freezing the advancement of time to the expiration date.

These extensions were clearly driven by the industry and its interests, and add nothing to the very purpose of copyright, which is to incentivise and reward creators for making new work.

Pollock calculated the optimal term of copyright to be around 15 years, substantially shorter than any current copyright term, which implies that existing terms are too long.^{174 175}

- **Copyrights and patents are nowadays grouped under intellectual property rights, despite the matter at hand being fundamentally different from tangible property.**

Exclusive property rights make sense for physical property because of their scarce and rival nature: tangible assets are limited in supply, so each one has one user and one owner. On the contrary, there is no limit to the supply of digital information unless we deliberately restrict it. In section 7.1.2 of the report we discuss how FOSS can be considered a public good, since it can be copied and used by anyone without affecting its accessibility or availability to

¹⁷¹ Book 'The Open Revolution', by Rufus Pollock (<https://openrevolution.net/>), published under the Creative Commons Attribution ShareAlike licence v4 (BY-SA)

¹⁷² Interview and mail exchanges with Rufus Pollock in February–July 2021

¹⁷³ <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32011L0077>

¹⁷⁴ Forever Minus a Day? Calculating Optimal Copyright Term, by Rufus Pollock, University of Cambridge, June 15, 2009. Source: https://web.archive.org/web/20130112152530/http://rufuspollock.org/economics/papers/optimal_copyright_term.pdf

¹⁷⁵ In her 2014 paper 'Copyright and Inequality', Lea Shaver, Professor of Law at the Indiana University Robert H. McKinney School of Law, argues that current (too-strong) copyright protection disturbs free markets. It makes books too expensive for people in developing countries, and even for poor people and linguistic minorities in industrialised countries, thereby increasing social inequality. Source: https://openscholarship.wustl.edu/law_lawreview/vol92/iss1/7/.

others. In a market system for information without monopoly rights, this limitless supply would result in prices tending to practically zero.¹⁷⁶

Of course, the industry has every interest in maintaining the idea of information and knowledge as property (which is a much more palatable concept than monopoly),¹⁷⁷ and keeping their exclusive right to make the copies. We should not forget, however, that copyright is one way to pay for the first instance of a work, which indeed can be hugely expensive to create. But there are other ways to fund innovation, one of which is to replace copyrights and patents with remuneration rights (as discussed below), preserving the incentive to create and innovate but without creating monopolies.

- **There is an overwhelming (and growing) dominance of big tech companies (not just in technology but in society as a whole)**. Pollock identifies three phenomena that have allowed these companies to become such extraordinary concentrations of power and wealth:
 - platform effects¹⁷⁸ (excluding rivals and inhibiting new entrants),
 - cost-free digital copying (allowing infinite economies of scale, making them unprecedented profits), and
 - intellectual property rights (giving them exclusive control of the software and algorithms that power their products and platforms).
- The WTO Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS) became effective in 1995 – not coincidentally shortly after the broad rise of the Internet – and required the implementation and enforcement of intellectual property rights by all WTO members.

Pollock calls this agreement a "treaty dedicated to reducing the flow of information", instigated by developed countries to protect and benefit their information-based industries, from software to pharmaceuticals. For where physical goods can be stopped at a border, it is almost impossible to erect technical barriers to the flow of information.

In addition, concentrated interests, usually corporate but also of very wealthy individuals, are better able to handle (legal) complexity than the rest of the world, helping to centralise power ever further. The same is true specifically for digital technology, which is complex, fast-moving and fundamentally abstract. The issues involved require value judgments, which subtly but significantly change the distribution of money and influence between media conglomerates, artists and the general public. Regulations are decided on a supra-national basis, reducing external

¹⁷⁶ As discussed in section 7.1.3.1 of the Report, Dries Buytaert makes a plea for FOSS to be considered a common rather than a public good. More generally: while copying and using information can be done at negligible cost, the total value that can be created from it (commercially or for society) is limited. That means that there is a limit to the amount of resources available to create and maintain the information. Free-riders extracting too much value from this pool of potential reciprocal resources pose an existential threat to the sustainability of the information at hand.

¹⁷⁷ Pollock calls this deliberate creation of confusion by industry "rhetorical hijacking". Richard Stallman calls 'intellectual property' a "*distorting and confusing term which did not become common by accident. Companies that gain from the confusion promoted it.*" "*It suggests thinking about copyright, patents and trademarks by analogy with property rights for physical objects.*" Source: <https://www.gnu.org/philosophy/not-ipr.html>.

¹⁷⁸ What Pollock calls platform effects here, we call network effects. These are already identified in section 4.5.7.2 of the report as an important pillar underneath the platform services of large Internet conglomerates.

scrutiny and leaving law-making to technocrats and corporate lobbyists with their special interests.

According to Pollock, unjust international monopolies currently affect free enterprise, free markets and free choice. The dead hand of monopoly artificially raises pricing or denies access altogether, thereby holding back and distorting innovation, resulting in stunted growth and competition, and lost opportunities (a market failure called the "tragedy of the anticommons").

Because of its non-excludable and non-rival nature, the natural way to treat information is as what Pollock calls a 'collective commons'. Treating information as property and deliberately imposing the monopolies of copyrights and patents on it limits people's access artificially, inflates prices, and curtails the scope for third parties to reuse the information in their own work. Free sharing is the only way for information to realise its full benefits for society as a whole.¹⁷⁹

C.3.2 An Open world

Pollock advocates an Open world in which all non-private information and knowledge, i.e. information that could be legally or legitimately sold or transferred to any third party, is freely available to all. That would allow anyone to use, build on (reuse) and share everything from statistics and research to newspaper stories, books and maps, from software and films to music, design and medical formulae.

The Open Definition,¹⁸⁰ created by the Open Knowledge Foundation and derived from the Open Source Definition (discussed in section C.1.2) in 2005,¹⁸¹ can be summarised as follows:¹⁸² Open data and content can be freely used, modified, and shared by anyone for any purpose.

In addition, three provisions can apply to the information at hand:

- attribution (to the creator/contributors),
- integrity (of the work),¹⁸³ and
- an insistence that shared derivatives must be shared alike (i.e. reciprocity).

According to Pollock, opening up all digital information and knowledge – and making all patented and copyright materials freely available – would solve the problem of the monopolies of information power, promoting competition, providing transparency and increasing the possibilities and incentives for innovation.

Doing away with copyrights and patents would also allow us to pay innovators and creators more, more fairly, and in more socially beneficial ways, using market-driven remuneration rights in place of intellectual property monopoly rights.

¹⁷⁹ Economic historian Eckhard Höffner has published extensively on the history of copyright, comparing Germany and the UK. He argues that the lack of copyright (enforcement) in Germany at the time, and the massive proliferation of scientific and practical books, academic papers, and hence knowledge, was responsible for Germany's rapid industrial expansion in the 19th century and its current industrial power. Source: <https://www.spiegel.de/international/zeitgeist/no-copyright-law-the-real-reason-for-germany-s-industrial-expansion-a-710976-amp.html>.

¹⁸⁰ <https://opendefinition.org/>

¹⁸¹ <https://opendefinition.org/history/>

¹⁸² <https://opendefinition.org/>

¹⁸³ The protection of the integrity of creative work is not part of the Free Software Definition and the Open Source Definition, since it is at odds with their clauses on non-discrimination against specific uses and/or users (as discussed in section [C.1.9](#)).

C.3.3 Remuneration rights as an incentive for innovations

Copyrights and patents are not the only way to pay for that first, expensive original. Pollock proposes the use of remuneration rights to offer creators and investors in this Open world a means of profiting from their efforts and risk-taking, and so give an incentive for further production, without granting exclusive rights.

The idea is to let market mechanisms between users and suppliers – neutral and democratically overseen – do their work in providing opportunities for everyone, and specifically to reward innovations according to take-up. Government would leave the market and entrepreneurs to decide what information is created and consumed, but would coordinate the raising and distributing of money. The first can be done through taxes, the latter through a fund. Demand-driven market mechanisms can be used to allocate all or part of the money collected, based on remuneration rights given to innovators and creators according to the value that their works create. This would optimise the outcome for society as a whole, including users and creators.

The benefits of such an alternative compensation system include not only business opportunities no longer wasted, but also the gains from reducing or eliminating costs in the current inefficient system, and benefits related to related to creativity and cultural freedom.

C.3.4 Some examples

Pollock discusses the Internet, public broadcasting, medical research, genetic science and recorded music as examples of ecosystems that are already (partially) publicly funded or using remuneration rights.

- The Internet, based on open technology, and not owned or controlled by any one firm, has undoubtedly created tremendous value. Its start and further development were mostly funded by governments through research funds. This in contrast to the traditional telecommunication networks.

Yet Pollock warns how the Internet is threatened by closed platforms (i.e. proprietary layers on top of internet) such as Facebook and Google, which have near-monopoly power in social media and infrastructural services. This results in negative effects in the form of companies and innovations that never made it, or which have been absorbed into these companies and neutralised, diminishing innovation as they disappear.

- The music industry provides several examples of the use of remuneration rights. Radio stations, for example, have blanket licences from collecting societies, which permit them to broadcast music without obtaining a licence for each piece they play. The collecting societies then divide that pot of money amongst the copyright holders, roughly in proportion to airtime.

Spotify is the modern quasi-open (platform) variant of this concept.¹⁸⁴ It provides audio streams in a freemium model, charging users for options such as commercial-free listening and downloading music. The subscription model increases the chances of customers extending their memberships (or they will lose their playlists and downloads). Royalties are paid under bulk licences from the rights holders, based on the relative number of streams.

Open platforms with competition may be great for users and artists, but a monopoly platform is more attractive to investors. That is why Spotify – despite

¹⁸⁴ <https://imedproject.org/2018/02/16/spotify--remuneration-rights-similarities-between-two-open-information-models/>

being quasi-open – is a developing monopoly in commercial hands and lacks universal access. The company has strong incentives to use its power to shape the development of the ecosystem in ways that preserve and enhance its grip. It will want to restrict or kill off innovations or developments that threaten its monopoly – a monopoly which if unchecked will give it power not only over music-listening but over artists and record labels, and over technological innovation related to music access and discovery. Moreover, it will never be in Spotify's commercial interest to price at a level that gives access to everyone, because users differ in their willingness to pay for the service.

Payment for these types of services could also be done through:

- taxes;
- a fee on top of your internet/mobile subscription;
- a levy on digital devices that play or store music; and
- taxes on online advertising: advertisements are the main source of income for most of these large platform providers that are based on content freely available from others, e.g. YouTube, Internet search. Other platforms, such as Facebook and other social media, have their content produced mainly by users themselves, while only a tiny proportion of this money is paid to the creators.

Money would then be distributed according to usage through remuneration rights fees, administered by collecting societies. As shown above, legal and administrative frameworks for this are already in general use in the industry. But many variations are possible, for example:

- a progressive scheme, in order to support the up-and-coming and experimental; and
- part of the money can be used for:
 - grants (compare current subsidies allocated up-front to particular artists or organisations to create new pieces and recordings);
 - awards (e.g. active consumer choice in the allocation of funding to particular artists, projects or even general policies); and
 - crowd sourcing (e.g. a form of up-front funding where artists would propose projects, such as an album or new song, with a budget, and the public would be able to allocate money through voting).

Reuse – which is frequent and important in music – fits naturally within the Open framework. Anyone would be free to build upon the work of others, but would then be liable to pay a proportion of their own remuneration rights payments (or other revenues) to those whose work they reused.

In an Open music model, government would not operate a streaming service or be involved in the delivery in any way. All it would do is establish a standardised, automatic, blanket-licensing regime to create a competitive market of music service providers. These providers would be mediating as technical distributors only, not as legal distributors. And anyone could create a new business or a new kind of business related to music.

- A similar story can be told about public service broadcasters and Netflix, an over-the-top (OTT) content platform and production company. The company has a huge stock market valuation based on its potential to become overwhelmingly dominant.
- Formulating and testing new drugs is extremely expensive, which is why the research and development of new medicines are supported through patents. Without the anticipation of high return, companies and their investors would never risk the expense of research, which would leave us with many fewer drugs rather than high-priced drugs.

Pollock argues that we could fund the development of new medicines in Open-compatible ways that are more effective than patents at rewarding innovators

and stimulating innovation. This would remove the deadweight loss caused by over-pricing.

Pollocks points out that almost every innovative medicine we now have started with work in a government-funded research lab – and that many of them were completed there too. Almost half of all medical R&D in the world today is funded directly by governments, and in basic scientific research (often very long-term and high-risk) the proportion is much higher.

He uses as an example the anti-HIV/AIDS drug AZT, for which Burroughs Wellcome (now part of GlaxoSmithKline, GSK) had merely carried out the final set of clinical trials to win approval for use from the US Food and Drug Administration (FDA).

Under the Open system, Burroughs Wellcome would not have been handed such an enormously valuable patent on the basis of work done, in large proportion, by others, including researchers paid for by taxpayers. Instead, manufacturing would be unrestricted and competitive, and this would keep prices close to the cost of manufacture, just like generic medicines today. Open access to the information would also encourage more scientists to work at the cutting edge of knowledge, to tackle diseases and disabilities more quickly.

The companies would continue to be rewarded for their work, because instead of patenting their innovations, they would apply for remuneration rights, which would entitle them to payments from a central fund in proportion to the health benefits of innovative drugs (e.g. based on quality-adjusted life-years, QALYs) – regardless of who actually manufactured them. This system creates incentives for targeted and socially beneficial research, while at the same time the lower prices of drugs gives patients dramatically expanded access to treatment.¹⁸⁵

Since research is a cumulative process and innovations that yield new medical treatments usually build upon and incorporate previous work, handling the re-use of information that is itself covered by remuneration is an important part of the Open model. Under present patent law, there are means for ensuring that such derivatives must license from the originator together with a dispute resolution system. Without a licence, reusers are liable for damages caused by infringing the patent. Under the system of remuneration rights, the lack of a licence would not prevent another party from building on existing work, although it would run the risk that later arbitration might award much of its future income to the original creators.

To prevent countries from freeloading, international agreements under which countries commit themselves to minimum levels of medical research funding would be required (as members of NATO, for instance, do currently in the case of defence spending). Likely, the percentage would differ between countries, with richer countries committing themselves to higher proportions. Countries might also agree to reciprocal recognition of remuneration rights, so that a remuneration right registered in one country would also be rewarded from the remuneration rights funds of other countries where a drug had saved lives or reduced suffering.

¹⁸⁵For example, the Open Insulin Foundation (OIF) consists of a group of biohackers "developing the first practical, small-scale, community-centered model for insulin production to make insulin accessible to all". Source: <https://hackaday.com/2021/08/23/open-source-insulin-biohackers-aiming-for-distributed-production/>.

C.3.5 The Open model

According to Pollock, the principal advantages of the Open model are:

- universal access to information;
- increasing innovation and creativity;
- maximising positive use of the capacities of information technology;
- increasing competition;
- ending of global monopolies over various forms of information;
- reducing inequalities of opportunities and outcomes; and
- increased global wealth.

The funding of valuable information will have the same sources under the Open model as it has now under the Closed model:

- business (e.g. journalism, film production, market research, advertising, fashion);
- sponsorship (whether commercial or pro bono);
- philanthropy (research, the arts, architecture, prizes, etc.);
- crowdfunding; and
- state spending (e.g. universities, learned societies, charities).

A good deal of the information produced this way is already Open – or could be. That goes, for example, for publicly funded research (as discussed in sections 2.3 of the report and A.1.1.3 and A.1.1.12 of Annex A). According to Pollock, the more information we can produce without direct state control the better, because this minimises politicisation and bureaucratisation and permits the greatest freedom for enterprise.

According to Pollock, replacing intellectual property monopolies such as patents and copyrights with remuneration rights is entirely compatible with continuing other means of encouraging innovation, and can enhance them. Philanthropists could for instance donate money directly to a specific remuneration rights pool. And private support for Open innovation could be encouraged by tax breaks.

Pollock argues that many of the technical and political aspects and infrastructures to implement remuneration rights as a viable alternative to the patent and copyright systems already exist and could be reused. Technically, this includes means of measuring value, defining ownership of innovations, and specifying what happens when innovations are built upon by others. Politically, this includes international legislation, means of arbitration, governing bodies for funds and means of securing sustainable funding.

Just like Bruce Perens (in section <C.2.6>), Pollock mentions the correct attribution of innovations as a crucial part of the remuneration system. Follow-on innovators are required to pay a proportion of their own remuneration rights payments to those whose work they built upon. Contrary to the present monopoly rights system, earlier innovators would not have an absolute right to prohibit reuse; they would have the right only to equitable remuneration. This way, holders of remuneration rights would be paid in accordance with usage and the value created by their innovations, and distributions would be made by transparent, pre-defined algorithms overseen by an independent assessors.

The proportionate allocation of money among the remunerations funds, however, would pose a problem. Since there is no limit to the supply of digital information unless we deliberately restrict it, the prices of digital information tend to zero in a market system without monopoly rights. As discussed in section 7.1.5, while digital

information may have significant one-off costs (i.e. the fixed cost), the cost of copying and distribution for each additional user (i.e. the marginal cost) is negligible. While traditionally market prices allocate spending and production in line with actual value and costs, in the new model prices fall as use rises (although the value is not falling). That leaves us without a market mechanism to determine the relative values of different types of information.

Note, however, that true market pricing of information goods is impossible at a fundamental level. Only by creating artificial monopolies can prices be attached to information, but still the producer rather than consumer demand sets the price, and the uniqueness (i.e. the non-commodity character) of information prevents true competition. In addition, it is politics that determines the length and nature of the monopoly rights that society grants to particular kinds of information (i.e. information realms), for example how much to spend on scientific research. So the problem of (and solution direction to) assessing value and allocating funds between different types of information is already present in the current system.¹⁸⁶

C.3.6 Implementation of the Open model

In the remainder of his book Pollock presents some suggestions on how to introduce and implement the Open model, and how the Open information movement – which started with the work of Richard Stallman, his Free Software Foundation, and the Electronic Frontier Foundation (EFF) and the Foundation for a Free Information Infrastructure (FFII) – should be organised to take on big tech and vested interests, and set change in motion.

The iMed project,¹⁸⁷ short for Innovating Medical Entrepreneurship and Delivery, has worked out how to replace the current patent-driven system for medical innovation by a system based on remuneration rights as described above. The main goal of this initiative is to provide access to vital medicines to those unable to afford them, and to drive the development of critical drugs for neglected and emerging infectious diseases.¹⁸⁸

Pollock is one of the partners in the iMed project,¹⁸⁹ which is financially supported by the Rockefeller Foundation.¹⁹⁰ You can find the details of the project in its white paper¹⁹¹ and FAQ.¹⁹²

¹⁸⁶ Chad Whitacre, the founder of donation platform Gratipay (later forked into Liberapay, <https://liberapay.com/>), once made a quick calculation on what would be required (from industry) to make FOSS development sustainable. He came to an amount of USD 2,143 per year per technical employee (at any company). Source: <https://gratipay.news/your-company-should-probably-pay-2000-per-person-for-open-source-9205443e209d>.

¹⁸⁷ <https://imedproject.org/>

¹⁸⁸ <https://imedproject.org/about/>

¹⁸⁹ <https://imedproject.org/who-we-are/>

¹⁹⁰ <https://imedproject.org/2017/11/01/imed-receives-rockefeller-foundation-funding/>

¹⁹¹ <https://imedproject.org/white-paper/>

¹⁹² <https://imedproject.org/faq/>