Licensing Framework for Choosing the Degree of Openness of a Process, Product or Service Based on Constraints of the Business Model

Project ZOOOM - 3Os and IP Awareness raising for collaborative ecosystems

Grant Agreement No. 101007385

Funded by the European Union

Document Control Information

Settings	Value
Document Title:	D1.2 – Licensing Framework
Project:	ZOOOM
Document Version:	16
Dissemination Level:	PA
Date:	29/09/2023

Document Author(s)

Organisation	Name(s)
KUL	Ivo Emanuilov
VTT	Jutta Suksi, Pasi Pussinen, Maija Federley, Emma Peltonen
UNITN	Davide Serpico
FSFE	Niharika Singhal, Lucas Lasota, Gabriel Ku Wei Bin
UP	Elizabeta Zirnstein

List of Abbreviations

Acronym	Explanation
3Os	Free and open source software, open hardware and open data (also referred to open assets as a general term encompassing all 3Os)
3D	Three dimensional
4Es	The following four emerging technologies: (1) artificial intelligence (2) quantum technologies and Internet of Trust (3) blockchain (4) robotics
ACM	Association for Computing Machinery
AGPL	Affero General Public License
AI	Artificial intelligence
AIA	Artificial Intelligence Act
API	Application Programming Interface
APP	Agence de Protection des Programmes
ART	Adversarial Robustness Toolbox
ASIC	Application-Specific Integrated Circuits
BLOOM	BigScienceLarge Open-Science Open-Access Multilingual Language Model
BoM	Bill of Materials

BSD	Berkeley Software Distribution (a Unix-like operating system)
BSL	Business Source License
CAD	Computer-aided design
CC	Creative Commons
CC-BY-4.0	Creative Commons Attribution 4.0 International license
CC-BY-SA (4.0)	Creative Commons Attribution-ShareAlike 4.0 International license
CC0	Creative Commons Public Domain Dedication
CCPA	California Consumer Privacy Act
CDDL	Common Development and Distribution Licence
CERN	The European Organization for Nuclear Research
CLA	Contributor Licence Agreement
CNRS	French National Institute for Scientific Research
COMPAS	Correctional Offender Management Profiling for Alternative Sanctions
CRA	Cyber Resilient Act
DCO	Developer Certificate of Origin
DMA	Digital Markets Act

DMCA	Digital Millennium Copyright Act
DSA	Digital Services Act
EC	European Commission
EDP	Eclipse Development Process
eGPL	Exception General Public License
EU	European Union
EUPL	European Free/Open Source Software (F/OSS) licence
FAQ	Frequently Asked Questions
FDL	Free Documentation Licence
FLOSS	The Free Libre / Open Source Software
FOSS	Free/Open Source Software
FPGA	Field Programmable Gate Arrays
FRAND	Fair, reasonable and non-discriminatory
FS	Free Software
FSF	Free Software Foundation
FSFE	Free Software Foundation Europe
GDPR	General Data Protection Regulation
GENCI	Grand equipment national de calcul intensif

GNU	A UNIX-like computer operating system that is free software and contains no UNIX code
GPL	General Public License
GPLv2	Version 2 of GPL
GPLv3	Version 3 of GPL
GPU	Graphics Processing Unit
HDL	Hardware Description Languages
IBM	International Business Machines Corporation
ICT	Information and communication technology
IDRIS	Institute for Development and Resources in Intensive Scientific Computing
IP	Intellectual property
IT	Information technology
JLA	Joinup Licensing Assistant
KPI	Key Performance Indicator
LGPL	Lesser General Public License
MIT	Massachusetts Institute of Technology
ML	Machine learning
MPL	Mozilla Public License
NGO	Non-governmental organization

NNW	Neural Network Weights
OD	Open Data
ODB	Open Data Barometer
ODbL-1.0	Open Data Commons Open Database License
ODC-By-1.0	Open Data Commons Attribution License
ODE	Open Data Ecosystems
ODM	Open Data Maturity
OES	The Organization for Ethical Source
OH	Open Hardware
OHL	Open Hardware License
OHL-P	Open Hardware License, Permissive
OHL-S	Open Hardware License, Strongly Reciprocal
OHL-W	Open Hardware License, Weakly Reciprocal
OKFN	Open Knowledge Foundation
OPN	Open Patent Non-Assertation
OS	Open Software
OSD	Open Source Definition
OSH	Open Source Hardware
OSHWA	Open Source Hardware Association

OSI	Open Source Initiative
OSPO	Open Source Program Office
OSS	Open Source Software
OSSg2	Second-generation Open Source
PCB	Printed Circuit Board
PDDL-1.0	Open Data Commons Public Domain Dedication and License
PLD	Product Liability Directive
R&D	Research and development
RAIL	Responsible AI Licenses
RAIL-M	Responsible AI Model License
REUSE	A project that attempts to standardise licence and compliance information in Free Software project repositories by utilising the SPDX specification
ROM	Read-only memory
RTL	Register Transfer Level
SaaS	Software as a Service
SBOM	Software Bill Of Materials
SCP	Structural, conduct-related and performance-related factors.
SME	Small and medium enterprise
SoC	System on Chip
SPDX	Software Package Data Exchnge

SSPL	Server-Side Public License
SyRI	System Risico Indicatie
TAPR OHL	Tucson Amateur Packet Radio Open Hardware Licence
TDM	Text-and-data mining
TODO	Open community of practitioners who aims to create and share knowledge about Open Source Program Offices or similar Open Source initiatives
TTO	Tech Transfer Office
UIA	Unemployment Insurance Agency
UK	United Kingdom
UNSAM	Instituto de Nanosistemas
VHDL	Very High-Speed Integrated Circuit Hardware Description Language
WP	Work package
ZOOOM	'3Os and IP awareness raising for collaborative ecosystems' (ZOOOM) Project, Grant Agreement No. 101007385

List of Figures

List of Tables

Table of Contents

Executive summary

The ZOOOM Licensing framework represents a tool for strategic decision-making. It is a practical guide for stakeholders about the variables and questions that one needs to consider in the adoption/use of open assets and technologies for business purposes, especially in innovation ecosystems. The framework investigates the intersection between legal and business aspects that can impact the decisions and conduct (in one word, the “journey”) of a company at different stages of development, depending also on its structural characteristics and internal organisation.

The ZOOOM Licensing framework consists of a set of intertwined principles and good practices that aim to facilitate the adoption of the principles of open source to the research, development and deployment of software, hardware and data driven technologies and business models. The framework helps knowledge creators and supporting organisations to identify the key elements of a business model, find their role in the ecosystem and choose one licensing model over the other.

The ZOOOM Licensing framework builds on the SCP paradigm that structures the framework under components relating to structural (S), conduct-related (C) and performance-related (P) factors. The traditional SCP paradigm is developed further to an extended SCP paradigm that gives a frame for open assets / open IP, i.e., FOSS, OH and OD.

The focus of the ZOOOM Licensing framework is in the following conduct-related (C) legal aspects:

1. IP strategy: eg, complementarities between open source, patents and trademarks
2. Licensing strategy: eg, outbound licensing
3. Licence management: eg, inbound licensing
4. Legal implications of hybrid assets

Each open asset – open source software (FOSS or OSS), open hardware (OH) and open data (OD), together 3Os – is first assessed from the perspective of its intrinsic features: ie, what makes it different from the other types of assets. Software revolves around code and copyright, whereas hardware adds a layer of design and physical elements that lie on a spectrum between ‘softwareness’ and ‘hardwareness’, and data adds additional value creation mechanisms due to its nature as non-rival good that often even lacks human-readable form. Each open asset is then analysed from the perspective of business, legal and social aspects.

One of ZOOOM's four emerging technologies (4Es), namely AI, is analysed deeper. We highlight the trends, regulations, and ethics; challenges and opportunities; copyright violations and litigations; and finally assess the role FOSS and other open assets in AI as a hybrid technology.

In the final section, we make five propositions for the development of tools based on the ZOOOM licensing framework:

1. Best practices checklist
2. Guide to choosing a licensing model
3. Standardising licence and compliance information
4. Using business profiles to address 3Os
5. Learnings from the Open Source Program Offices (OSPOs)

These propositions will be taken into account in the next phase of as part of the tools to be provided by ZOOOM to the companies and other stakeholders involved in 3Os.

1. Introduction

"The intellectual property situation is bad and getting worse. To be a programmer, it requires that you understand as much law as you do technology"

Eric Allman[1]

The ZOOOM Licensing framework represents a tool for strategic decision-making. It is a practical guide for stakeholders about the variables and questions that one needs to consider in the adoption/use of open assets and technologies for business purposes, especially in innovation ecosystems. The framework investigates the intersection between legal and business aspects that can impact the decisions and conduct (in one word, the "journey") of a company at different stages of development, depending also on its structural characteristics and internal organisation.

These are the central questions that the framework aims to address:

- How to create value from open source software,[2] open hardware, and open data?
- How to manage IP given the business aims of a company, in relation to both the assets and technologies developed by others (eg, compatibility issues and legal compliance/sustainability) and the company's own products?
- What are the main motivations for using and developing open assets, apart from financial or strategic ones?
- What are the challenges, risks, and benefits of using and developing open assets?
- What are the interactions between FOSS, OD, and OH (eg, peculiarities, differences, and complementarities)?

The framework also provides an overview of how to combine licensing strategies with business models based on a variety of economic, legal, and social aspects driving a company's choices. In addition, it addresses structural parameters such as the type of company and its technological area and industry.

[1] Sendmail founder. See https://www.rocket.chat/blog/open-source-quotes

[2] The terms 'Free Software (FS), 'Free and Open Source Software' (FOSS) and 'Open Source Software' (OSS) are used interchangeably in this deliverable.

Looking at the ZOOOM Licensing framework in more details, it consists of a set of intertwined principles and good practices that aim to facilitate the adoption of the principles of open source to the research, development and deployment of software, hardware and data driven technologies and business models. The framework helps knowledge creators and supporting organisations to identify the key elements of a business model, find their role in the ecosystem and choose one licensing model over the other.

Our starting point is that business models based on open source as a development and licensing model are nuanced. They can be fairly easy or indeed very complex to understand. We take the pragmatic view that licensing models, like business models, evolve over time. They exist to serve the communities around specific open asset projects and the needs of businesses that adopt and contribute to these projects. To this end, we focus on both traditional open source licensing models (ie, copyleft, permissive, and combinations of the two) and more nuanced models which, strictly speaking, do not fit into the OSI Open Source Definition[3] or Open Knowledge Foundation (OKFN) Open Definition[4] but aim to ensure source (code or data) availability and modification rights to downstream users (e.g., the Elastic Licence, the Business Source Licence, the Server Side Public License, or the Cryptographic autonomy license). Furthermore, we also explore commercial open source licensing models which combine open source and proprietary licensing, as well as complementary strategies between open source and other exclusive IP regimes, such as patents and trademarks.

The ZOOOM Licensing framework builds on the SCP paradigm that structures the framework under components relating to structural (S), conduct-related (C) and performance-related (P) factors. The traditional SCP paradigm is developed further to an extended SCP paradigm that gives a frame for open assets / open IP, ie, FOSS, OH and OD. Figure 1 below shows the extended SCP paradigm and the main building blocks and components that a company needs to address when building business based on open technologies.

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

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

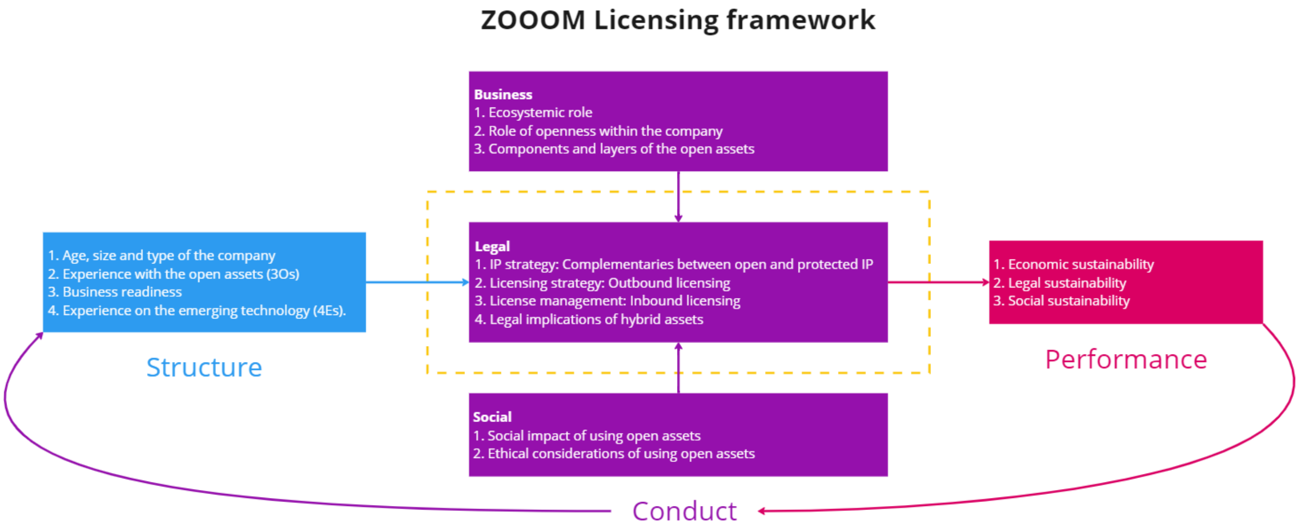

ZOOOM Licensing framework
Business
1. Ecosystemic role
2. Role of openness within the company
3. Components and layers of the open assets
1. Age, size and type of the company
2. Experience with the open assets (3Os)
3. Business readiness
4. Experience on the emerging technology (4Es).
Structure
Legal
1. IP strategy: Complementaries between open and protected IP
2. Licensing strategy: Outbound licensing
3. License management: Inbound licensing
4. Legal implications of hybrid assets
1. Economic sustainability
2. Legal sustainability
3. Social sustainability
Performance
Social
1. Social impact of using open assets
2. Ethical considerations of using open assets
Conduct

Figure 1. Visualisation of the ZOOOM Licensing framework based on the extended SCP paradigm with its main building blocks and components.

In a nutshell, the framework can be explained in the following way: The structure, business conduct and social conduct feed into the legal conduct of the company; the core of the ZOOOM licensing framework is at the centre of this interaction, as part of the legal conduct; and both areas, structure and conduct, feed into the performance part of the business of the company, which then affects and interacts with the conduct-related and structural parts.

In practical business terms, the framework can be used as a tool for instance when a company is thinking of broadening the use of open assets (FOSS, OH or OD) in its business.

The company would need to identify the structural elements in the market it enters (eg, what kind of competition exists), its experience with regard to the open assets (eg, how well does it understand the effects of copyleft licences) and its business readiness for the market (eg, technical maturity of the company). These aspects are collected under Structure in the blue box.

In addition, it needs to identify its own role in the ecosystems (eg, will it only use or also contribute to open assets) and specify technological components it uses (eg, how their technology stack looks like) to be able to make strategic decisions on the role openness takes in its business. These aspects are collected under Conduct-Business in the purple box.

In addition to business related decisions, the company needs to assess its position regarding social (eg, what kind of impact it aims to generate) and ethical (eg, which ethical aspects it prioritises over pure profit-making) aspects. These aspects are covered by Conduct-Social in the purple box.

All of the above decisions affect the Legal Conduct of the company, ie, its IP strategy, licensing strategy, licensing management and its decisions on the hybrid assets that combine FOSS, OH and OD. These are covered by Conduct-Legal in the purple box.

Finally, the above Structural and Conduct-related decisions and actions affect the Company's performance in terms of economic, legal and social sustainability (Performance in the red box) with a constant loop of feedback to Conduct and Structure.

In the following, we will address the extended SCP at a general level under Section 2 and per each open asset, ie, FOSS, OH and OD under Section 3.

2. Extended SCP as the basis for ZOOOM licensing framework

In this section, we propose a framework for the description of business structure and activities, which we will integrate and expand to include further aspects that are more closely related to the 3Os landscape.

We take inspiration from a classical paradigm in industry theory known as *Structure-Conduct-Performance* (SCP; for some reviews, see Lelissa & Kuhil 2018; Lipczynski et al. 2005, capp. 1 and 9). The central hypothesis of SCP is that structural characteristics of a market determine the behaviour (conduct) of firms within that market, and the behaviour of firms within a market determines measurable market performance (Bain, 1951, 1956; Mason (1939, 1949). SCP is often used to analyse and predict how different factors influence the behaviour and outcomes in various industries.

Although SCP rests on a variety of idealised assumptions as regards the relationship between structure, conduct, and performance, it can still be a useful tool to assess a company's characteristics and main business aims according to certain standard categories shown below (for a complete discussion, see Lipczynski et al. 2005, pp. 7-10):

Market Structure:

1. Number and size distribution of buyers and sellers

2. Entry and exit conditions (barriers)
3. Product differentiation
4. Vertical integration and diversification

Conduct:

1. Business objectives
2. Pricing policies
3. Product design, branding, advertising and marketing
4. Research and development
5. Collusion
6. Merger

Performance:

1. Profitability
2. Growth
3. Quality of products and service
4. Technological progress
5. Productive and allocative efficiency

The SCP paradigm suggests different causal relationships between structure, conduct, and performance:

- Structure → Conduct → Performance: This is the traditional direction of causality in the SCP. For instance, a more concentrated industry with limited competition might lead to firms engaging in collusive behaviour or setting high prices, which could negatively impact consumer welfare and overall industry performance.
- Performance → Conduct → Structure: In some cases, it's argued that industry performance can influence conduct and structure. For instance, if firms are in a competitive market experience declining profit, they might change their behaviour (conduct) by investing in research and development to innovate and differentiate their products. This change in conduct could then alter the industry structure.

Based on the SCP paradigm, we propose a comprehensive framework called "Extended SCP", which will serve as the theoretical basis of the ZOOOM licensing framework bridging the gaps between and integrating business and legal aspects relating to open assets, ie, FOSS, OH and OD. Differently from the classical SPC, which is intended to assess a company's conduct and performance based on a *market* *structure*, this framework focuses on the structural component mostly at the level of a single company, under the assumption that a company's internal organisation will impact its conduct and performance. Indeed, as mentioned in the Introduction, our aim is to guide companies about the variables and questions that one needs to consider in the adoption and use of open assets.

It should be emphasised that the original SCP paradigm constitutes a typical industrial organisation model, as it analyses and explains the relevant elements and dynamics of a business sector. From this overall vision, other scholars (Porter, 1980; Grant, 1991) have developed over time a stream of strategic models that offer prescriptive elements to explain and account for the behaviour of individual firms, aiming to expand the SCP paradigm into individual businesses. The Extended SCP follows such developments as it is intended as both *descriptive* and *prescriptive*: on one hand, it allows for the operationalization of various characteristics of a company; on the other hand, it suggests future choices based on the characteristics identified in the descriptive component.

The Extended SCP allows one to operationally define the various characteristics of a company that wants to engage with the 3Os. This approach seeks to adapt the SCP framework to the realm of open source software, open hardware, and open data. This requires delving into the internal organisational parameters of companies engaging with the 3Os while considering economic, legal, and social aspects of their conduct and performance. By expanding the SCP framework to include these dimensions, we can get a more holistic understanding of the intricate interplay between organisational attributes and 3Os engagement, with a special focus to economic, legal, and social dimensions.[5]

The structural attributes of companies, their legal strategy, economic motivation, and social commitments collectively shape their performance within the 3Os ecosystem. This Extended SCP analysis not only provides a comprehensive lens to understand companies' behaviour and outcomes but also underscores the need for a balanced approach that aligns economic interests with ethical and societal considerations. As

[5] Note that the classical SCP framework posits a stable relationship between its components (Church and Ware, 2000). However, in real world scenarios, there is arguably some degree of circularity and dynamism due to interactions among structure, conduct, and performance within an industry, also due to changes in the market environment (see Lipczynski et al. 2005, p. 7). Additionally, the SCP does not take into account the role of external factors such as technological changes, government policies, and consumer preferences, which can also significantly influence industry dynamics and, consequently, a company's choices. In order to facilitate our analysis, we shall mostly consider the relationship between structure, conduct, and performance in "static" terms, taking into account "crystallised patterns" such as the conduct of a company at a given developmental stage.

the landscape of open source software, open hardware, and open data continues to evolve, this framework offers a theoretical grounding to provide valuable insights for companies seeking to navigate this dynamic domain while ensuring sustainable and responsible engagement.

2.1. Extended SCP: Structure

The organisational parameters that govern a company's approach to the 3Os framework are multi-faceted. The SCP paradigm traditionally considers industry sector, business environment, market, company boundaries, and ecosystem as structural factors. In the context of the 3Os, the focus shifts to factors like a company size and type (start-up, SME, large enterprise) and its previous experience with open technologies (newcomer or expert user of the 3Os). The technological domain where the company operates becomes a defining characteristic, guiding its strategic alignment within the 3Os ecosystem. These attributes collectively shape the company's structure, positioning it within a dynamic environment of collaboration and competition.

Core aspects of a company’s description are:

1. Age, size and type of the company: start-up / SME / large company; new companies / mature companies; company /research institute
2. Experience on the open assets (3Os): type of open technology (FOSS, OH, and/or OD) that is needed given the business aims; and the experience of the company (newbie / basic user / expert user /contributor / initiator / focal firm);
3. Business readiness: knowledge on the markets (industry sector; company boundaries; business ecosystems; available financial resources;
4. Experience on the emerging technology (4Es: AI, Blockchain, Quantum, Robotics)

In the context of ZOOOM, structural parameters can have a significant impact on the conduct and performance of a company. For example, smaller enterprises or startups may have great advantages in terms of cost savings, enabling them to tap into a vast repository of ready-to-use solutions, thereby bypassing the resource-intensive path of developing products from scratch. This not only accelerates time-to-market but also empowers smaller players to compete effectively with their larger counterparts, removing the traditional resource barriers that may have once limited their ambitions.

Structural elements also define the potential. for instance. for smaller enterprises to disrupt the market practices of current market leaders.

The technological area and industry, too, can influence the choice of whether to engage or not with the 3Os and how. For example, the emerging field of robotics is extensively based on open source software and open hardware, and it can be difficult to catch up with the latest developments by relying only on proprietary solutions. If the value proposition of a company involves complex interactions with a network of other actors and organisations, such as in the area of additive manufacturing, open source assets can provide more compatibility which is crucial for the success of collective efforts.

2.2. Extended SCP: Conduct

Companies can engage in the 3Os in many ways (eg, as users or makers/contributors of open assets, or both) and for different reasons. In many cases, strategic and competitive advantages are at the core of a company's decisions. However, the motivation behind a company's engagement in the 3Os is not necessarily economic in nature. For instance, they can involve the potential for growth (both personal and collective), technological innovation, practical needs, psychological motivations (eg, freedom, the potential for career development, self-enjoyment, ideology), and social and ethical values (eg, reciprocity, altruism, democratisation of knowledge).

Understanding what motivations can lead to the adoption of the 3Os requires reference to the many potential aims of a company, including not only financial goals but also *values* that companies may prioritise based on different considerations. This can impact the licensing choices of a company, too. Classical work on Corporate Social Responsibility (Carroll 1991) sketches a hierarchy of priorities that can be useful to frame the discussion:

- Economic responsibility: maximising profits;
- Legal responsibility: compliance with legal obligations;
- Ethical responsibility: meeting new values and movements, e.g., public good, environmental sustainability, and social sustainability;[6]

[6] See the case of B Corporation. The B Lab Europe is part of an international network that creates economic systems change through standards, policies, tools, and programs for businesses, and companies that are leading the way. B Lab provides certificates to companies that meet high standards of social and environmental performance, transparency, and accountability (https://bcorporation.eu/).

- Philanthropic responsibility, which is discretionary or voluntary.

Our framework attempts to reflect this classical hierarchy. By analysing *conduct* and *performance* of a company in terms of three core components: business aspects (containing economic responsibility), legal aspects (including legal responsibility), and social aspects (note that social aspects include both ethical and philanthropic responsibility of Carroll's hierarchy). Let us briefly consider such three facets of *conduct*.

2.2.1. Business Conduct

Business Conduct involves the business-related advantages and motivation for engaging in the use or development of the 3Os, including pursuing competitive advantage, reduction of development costs, speed and efficiency, access to knowledge or assets, control over technological development, talent acquisition and retention, network effects, and interoperability. It also includes practical needs. Companies must define their future behaviour with an eye on the 3Os they aim to employ (FOSS, OH, OD) and on balancing between being a user and a contributor/maker, depending on their strategic goals and broader priorities, including social and community engagement.

Core aspects to be considered are:

1. Ecosystemic role: This includes the company's role in the ecosystem; value-creation and value-capture aspects in the multifaceted and multi-layered ecosystem; and company's engagement with the community. These depend on business aims, but also on other types of priorities (e.g., social aspects and engagement with communities), which can also determine a company's role in their ecosystem.
2. Role of openness within the company: This includes motivation to use open assets, and choice of role (users vs. makers vs. both). This can also include value-capture to be gained using open assets, such as the revenue model, value proposition, as well as niche considerations on pain and needs of customers that a company aims to meet.
3. Components and layers of the open assets: This includes a thorough analysis of the components used by the company and identification of different technological layers that are connected to the open assets. It also includes strategic decisions on open parts and closed parts of the components and issues relating to management of hybrid combinations of the open assets.

At the heart of many companies' open source engagement lie strategic and competitive considerations. Generally speaking, the main drive is the allure of cost savings and resource management, particularly for small and resource-constrained firms like start-ups.

First, standardisation, flexibility, and modularity represent critical aspects for companies, particularly in an era of rapid technological proliferation and increasing complexity. Open source technologies provide a platform for establishing and promoting industry standards, facilitating interoperability and compatibility among diverse systems. The advent of standardisation, such as APIs (Application Programming Interfaces) and reference implementations, has become increasingly important, particularly for smaller players seeking to navigate complex ecosystems. Furthermore, participation in open-source development compels companies to uphold international standards, ensuring that the software, data, or hardware they produce adheres to recognized quality benchmarks. Commitment to open source not only enhances the credibility of the company but also results in a more consistent and homogeneous user experience. End-users, in fact, can benefit from products that adhere to established norms, ensuring compatibility, stability, and reliability.

Moreover, open source solutions, characterised by flexibility and modularity, are well-suited for customisation. Companies can adapt these solutions to their specific needs, tailoring features, functionality, and interfaces to align with their strategic objectives. This adaptability empowers organisations to create differentiated offerings while maintaining compatibility with established industry standards, allowing for seamless integration with other products and services.

Another appealing aspect is that open source assets offer a solution to the challenge of vendor lock-in for both companies and customers, namely, the dependence on a single vendor's proprietary solutions, which can limit innovation and flexibility. Open source combats these issues by promoting interoperability. On one hand, by incorporating open source software, data, and hardware, businesses avoid relying solely on one vendor: they can customise and extend these technologies, ensuring alignment with their unique needs. On the other hand, customers benefit from open source by accessing a competitive marketplace and enjoy interoperability and competitive solutions: multiple vendors offer compatible solutions, driving innovation and preventing monopolistic practices. In this sense, customers can easily switch vendors or integrate third-party offerings, ensuring uninterrupted services.

As regards a company’s role, companies can play different roles in their industry and ecosystem. For example, *users* tend to adopt open-source elements developed by others. What we call *contributors* or *makers*, on the other hand, usually have an active role in the development of open technologies, actively engage with one or more communities, and are thus involved in innovation processes that revolve around the 3Os. For instance, they can contribute to an open-source platform, share their projects online with global communities, take on the role of system integrator or an OSS framework leader, or make minor customizations to an already existing product (Browder, Aldrich & Bradley 2019; Li et al. 2020; Troxler & Wolf 2017).

Being involved in open-source ecosystems brings several benefits to contemporary businesses. Here, the creation of new knowledge is facilitated by joint research work, collaboration, expertise sharing, and the development of a common knowledge base to which communities of developers, too, contribute. Ecosystems growing around the 3Os can be recognised as knowledge ecosystems, where knowledge sharing and knowledge creation are central activities (see Koening 2012). This connects to the potential for innovation, where the 3Os play a vital role in business operations, encouraging collaboration and in many cases providing competitive advantage. Open-source components are indispensable also in research and development, particularly in cloud architecture where reliance on major providers is common. Open libraries, software and tools enhance efficiency and cost-effectiveness, enabling smaller companies to compete effectively. Customised solutions, characterised by flexibility and modularity, are employed to meet specific client requirements while safeguarding intellectual property.

As regards the role of open assets in companies’ businesses, the main element is FOSS, often serving as a foundation for data analysis and scalable solutions. FOSS related businesses can significantly support OD and OH initiatives by providing open tools and collaborating with their communities. Across diverse industries, FOSS adoption seems to be on the rise, contributing to precise analysis and informed decision-making. Monitoring the hybrid interaction between FOSS and other open assets is imperative for companies to unlock the full potential of open technologies and foster continuous innovation and growth. The integration of the 3Os enhances a company's portfolio and competitive advantages, with these domains complementing each other across various sectors.

Overall, the interplay between open and closed elements represents a strategic balance, with companies making informed decisions about leveraging these elements. The 3Os provide a robust foundation for innovation, cost savings and broader reach through collaboration within communities. Many companies actively contribute to the open-source community and balance collaboration and monetization by combining open-source and commercial licences. Collaboration within the open-source community strengthens code quality and supports business growth. While open technologies offer numerous advantages to businesses, they also pose some challenges, such as integration complexities and security concerns. Interactions between the 3Os are multifaceted, with each domain playing a distinct role and the potential for synergy when combined strategically.

2.2.2. Legal Conduct

Within the SCP framework, legal conduct involves the intricate legal choices and licensing strategies that companies must navigate when using or developing open assets for business purposes. These choices hold the potential to influence the extent of collaboration, the scale of innovation, and the alignment of interests among stakeholders.

Core aspects to be considered in the legal conduct of a company are:

1. IP strategy: complementarities between open source, patents and trademarks
2. Licensing strategy: Outbound licensing
3. Licence management: Inbound licensing
4. Legal implications of hybrid assets

2.2.2.1. IP strategy

The intellectual property (IP) strategy of a company is essential to its success. However, many companies struggle to create a coherent IP strategy because intellectual property is not homogeneous. IP consists of subject matter, or assets, which are governed by different legal regimes, such as patents, copyrights, trade marks, industrial designs, geographical indications, domain names or undisclosed information (eg, know-how or trade secrets). One common aspect is that the protection granted by these different regimes typically rests on the principle of exclusivity, ie, the proprietor can exclude any third party from making, using, selling, reproducing etc. the respective subject matter.

As discussed in D1.1,[7] the fundamental distinction between proprietary licensing models and open source licensing models lies in the modalities of these exclusive rights, ie, how they are exercised in practice. Open source uses the position of the

[7] Ivo Emanuilov and others, ‘ZOOOM Deliverable 1.1 Literature Review of Legal Cases in Free and Open Source Software, Open Hardware and Open Data’ (European Commission 2023).

rights holder as the ultimate proprietor to reverse the effects of exclusivity and ensure that the asset covered by exclusive rights can be subject to free, unfettered downstream distribution, use, modification and sharing. The licence grant always stems from the original holder of the exclusive rights which guarantees that downstream users always get legal certainty of the scope of the granted rights. This is known as the direct licensing model of open source. The model allows redistribution of modified or unmodified works but does not grant rights to grant further sub-licences. Furthermore, open source licences are not transferable because the licence is always granted from the original rights holder to all downstream recipients. A proper understanding of these differences between proprietary and open source licensing models is key to developing a strategy that combines both models.

A company typically has a diverse portfolio of intellectual property assets, ranging from trade marks, through copyright in audio-visual works and computer software, to patents in inventions. In a typical IP portfolio, these assets are often complementary. It is therefore essential that a company builds an inventory of its IP assets before making any decisions on how to transact with them. IP strategies come in all shapes and sizes and it is usually a good idea to retain the services of an IP professional to help devise one. That being said, most IP strategies would fall into one of these categories:

- **Closed IP strategy**. This is a strategy where the company relies entirely on the exclusive rights granted by the various IP regimes. The company engages in transactions with external parties only by means of limited non-exclusive licence grants with minimum royalty requirements, secrecy or access control measures, incl. technological protection measures. This is a typical approach for companies with patent portfolios which rely on a combination of patents and trade secrets.
- **Open IP strategy**. This is strategy where the company relies on the direct licensing model of open source and/or open patent non-assertion pledges[8] to promote inclusivity as opposed to exclusiveness. Hardly any company uses this strategy alone.

[8] Open patent non-assertion (OPN) pledges are a legal device that relies on the doctrine of estoppel to prevent those companies that seek the benefits of open source in their own businesses from launching attacks against open source products and platforms. Some of the key benefits of OPNs are (1) patent holders' ability to determine exactly which patents (and related technologies) they wish to pledge and offer the public transparency in the process, (2) defensive termination relative to a broader range of incoming patent attacks, and (3) non-assert promise and defensive use only terms that are designed to remain in force for the life of the patents, even if sold or transferred. See more in Google, 'Patents in the Service of Open Source' (*Google Patents Site*, 20 September 2023) <https://www.google.com/patents/opnpledge/> accessed 20 September 2023.

- **Mixed IP strategy**. This is a strategy where a company relies on a combination of open IP strategies for some elements of its core IP assets and exclusive protection and proprietary licensing for others. This is perhaps the most common approach in business today because it gives a company the flexibility to leverage the different tools offered by IP law depending on the needs. On the one hand, this allows the company to create open source communities around some of its projects, frameworks or components, and thus reap the benefits of collaborative development and community contributions.[9] On the other hand, keeping exclusivity in some of the assets creates monetisation opportunities by means of commercial licensing where the business needs dictate so.

The IP strategy does not exist in a vacuum because it is highly dependent on the company's business model and its evolution over time. For example, in recent years, a growing number of companies have taken the path of adapting their open IP strategies by relicensing their projects under restrictive terms for certain fields of use, such as cloud computing. Recent examples of this trend are MongoDB, Elastic, and HashiCorp all of which switched to source-available licences which impose restrictions when the software is offered as a service. While this is a clear departure from the principles of the Open Source Definition, which prohibits any field of use limitations and discrimination, these businesses have argued that such limitations are the only way for them to remain committed to the spirit of open source while still making money.[10] This hints at further differentiation within the open IP strategy model where a company may choose to distribute some of its assets under an open source licence and others under a source-available licence with certain field of use restrictions. Arguably, the main driver is to keep the communities active around the project by keeping the source code available, but still introduce curbs on cloud companies who may make money by offering the software as a service. There is of course the caveat that such actions may be perceived as hostile by the community around a project and

[9] Consider, for example, the recent release of the Llama 2 large language model by Meta under the Llama 2 Community Licence Agreement, which mimics some but not all conditions known from typical permissive open source licences. It is easy to discern the pattern of a company that wants to create a community around a product, incentivise collaborators to join the community, and reap the benefits of collaborative development for its internal operations.

[10] According to the press release issued by HashiCorp, such changes were necessitated by "vendors who take advantage of pure OSS models, and the community work on OSS projects, for their own commercial goals, without providing material contributions back. We don't believe this is in the spirit of open source." See Armon Dadgar, 'HashiCorp Adopts Business Source License' (*HashiCorp Blog*, 10 August 2023) <https://www.hashicorp.com/blog/hashicorp-adopts-business-source-license> accessed 10 September 2023.

could eventually lead to the open source variant being forked and developed independently.[11]

Regardless of the differences in these models, a successful IP strategy should focus mostly on the highly valued intellectual property. Van Lindberg highlights that most of the IP is in fact of the supporting variety, ie, assets that do not get licensed often and that can reduce the total value of the portfolio if companies keep them closed.[12] Indeed, companies should focus on the IP assets that differentiate their business offering from that of other businesses, and engage in open source collaboration over the assets that play a merely supporting role.

2.2.2.2. Licensing strategy

The outbound licensing strategy of a company is a function of its general IP strategy. It is, in fact, the world-facing side of that strategy. In other words, this is a statement to potential customers and contributors which sets the modalities for using, sharing or modifying a project, framework, library or other component released by a company.

It is considered best practice to have an **open source policy** that outlines, among other things, the company's licensing strategy.[13] Importantly, open source policies do not shield companies from claims because copyright infringement is tied to a strict liability regime where subjective elements, such as intent, do not play a role. However, it can facilitate collaboration between different teams across multiple countries by creating a common frame of reference. The complexity of an open source policy depends on the size of the company and the range of involvement with open assets.

In practical terms, an open source policy should cover at least the following topics[14]:

- Compliance processes, incl. inbound and outbound licence compatibility, vertical licence compatibility and provenance
- Code release and contributions

[11] Amazon's fork of Elasticsearch when Elastic decided to switch the project to the Server Side Public Licence is a recent example. See Carl Meadows and others, 'Stepping up for a Truly Open Source Elasticsearch | AWS Open Source Blog' (*AWS Open Source Blog*, 21 January 2021) <https://aws.amazon.com/blogs/opensource/stepping-up-for-a-truly-open-source-elasticsearch/> accessed 20 September 2023.

[12] *Patents & Open Source: Working with 'Mixed' IP Strategies* (Directed by Van Lindberg, 2020) <https://www.youtube.com/watch?v=R9UKUSj8BcQ> accessed 18 September 2023.

[13] Heather Meeker, *Open (Source) for Business: A Practical Guide to Open Source Software Licensing* (Third Edition, Independently published 2020) 171.Meeker 171.

[14] ibid 171–172.

- Baseline requirements for vendor contracts and other transactions

Most companies initially focus on compliance as the primary source of legal risks and only then gradually expand the policy to cover other topics.

An open source policy should be accompanied by adequate and manageable **business processes** and internal **allocation of roles**. For example, approvals for using strong copyleft licensed components in a project may be subject to prior approval and there should be an easy and preferably automated workflow to get this approval. Similarly, a company should have a clear and preferably standardised process for storing and processing licensing information about the software it uses or intends to use. Furthermore, companies may wish to adopt an approval process that is aligned with the intended release cycle so as to avoid delays. These processes can now be successfully automated using tooling such as FOSSA or Black Duck. Business processes for approval of requests to use open source components are often matched with appropriate roles within the company responsible for such approvals. These can be purely legal roles but more often than not it is an engineering role which combines a good understanding of the legal issues.

The **review processes** described in open source policies are typically conducted on a licence-by-licence basis. However, in some cases reviews may be conducted package-by-package, especially in cases of security concerns.[15] Some companies may even choose to create a repository (sandbox) of approved packages and prohibit, as a matter of policy, the use of packages that are not included in the sandbox. The review may be limited to licence compatibility or extended to cover patent infringement, export control or other legal matters.

Open source policies should be driven by the **use case**. Clearly, open source policies for companies that distribute open source software and hardware will differ significantly from a policy for a cloud service provider. In any case, even if the company believes that the product is not going to be distributed, it is a good idea to implement solid record keeping and licence information management practices. It is almost inevitable that a distribution will occur at some point. For example, this could happen where a private instance of a cloud solution is provided to a customer, or the software is distributed to another legal entity in the course of corporate restructuring, such as a merger or acquisition. Failure to furnish proof of good compliance management could have significant adverse consequences for future business opportunities.

[15] ibid 173.

An essential part of any licensing strategy is also a reasoned **statement about when and how the licensing model might change** and whether this would improve the revenue stream and/or strengthen the business model. There are three main paths to relicensing: (1) relicensing from proprietary to open source; (2) delayed open source publication or (3) relicensing from (i) more restrictive to more permissive open source licence, or (ii) from more permissive to more restrictive (open) source licence.

Relicensing from proprietary to open source is not that common, but we have some good examples, such as Blender, the popular 3D computer graphics software. Initially, Blender was developed as an application for internal company use only. When it first commercialised, Blender adopted a freemium strategy whereby a free version was available for downloading with the company selling product keys that unlock additional features. After a few initial successful fundraising campaigns with investors, the company behind Blender eventually failed and was taken by its investors, bringing all development efforts to a halt. The developer behind Blender decided to set up a non-profit entity, the Blender Foundation, with the intention of making Blender open source. Thanks to the community of more than 250,000 users, the foundation raised €110,000 in a few weeks, just about enough to get back Blender from the investors. In the wake of these events, Blender was officially released as open source under the strictest copyleft licence, the GPLv2.[16]

Delayed open source publication can be described as “the practice of publishing a software release under a proprietary license, then later publishing that release's source code under an open source license”[17]. Some examples of this approach include:[18]

- Aladdin Ghostscript
- Sleepycat and BerkeleyDB
- Sentry (Business Source Licence, BSL)
- Codecov (BSL)
- HashiCorp (BSL)
- CockroachDB (BSL)[19]

Relicensing within the family of (open) source licences. Typically, when projects decide to change their outbound licence, the change is almost always from a more

[16] Blender Foundation, ‘History’ (*blender.org*) <https://www.blender.org/about/history/> accessed 29 September 2023.

[17] Karl Fogel, ‘Delayed Open Source Publication -- Research’ (*GitLab*, 13 September 2023) <https://code.librehq.com/ots/dosp-research> accessed 23 September 2023.

[18] ibid.

[19] For each BSL release all associated alpha, beta, major, and minor (point) releases become Apache 2.0 on the same day three years after the major release date.

to less restrictive licence (eg, GPL to MIT), although there are also examples in the opposite direction.[20] The possibility of relicensing largely depends on the availability of a contribution agreement which would facilitate the process. The general recommendation for open-source-first businesses is to always start with a more restrictive licence and move to a more permissive one if it becomes obvious that it is a better fit.

Some companies, like Cockroach Labs, have adopted non-open source licences (BSL) for their major releases which automatically become Apache 2.0 released three years after the release date. Under the BSL, all features and the source code of CockroachDB are available, but users may not use it in a 'software as a service' configuration without an agreement with Cockroach Labs. This means that there is a guaranteed timeline for transition from source-available to open source licence for every major release.

2.2.2.3. Licence management

Licence management is part of the broader licensing strategy. It is the company-facing part of the IP strategy and focuses on ensuring horizontal compatibility between the inbound licences. The shorthand rules in the case of software are the following:[21]

- If there is a GPL licensed code in a program, it must all be provided under the GPL.
- LGPL licensed code should only be integrated into a program with other code as a dynamically linked library.

Heather Meeker provides a good illustration of the horizontal incompatibility issue in the case where the target software contains code covered by inbound terms under various licences, including the strong copyleft GPLv3. No outbound licence would work in this case:[22]

[20] See the example discussed above on companies relicensing projects under source-available licences, like the SSPL.
[21] Meeker (n 13) 70–71.
[22] ibid.

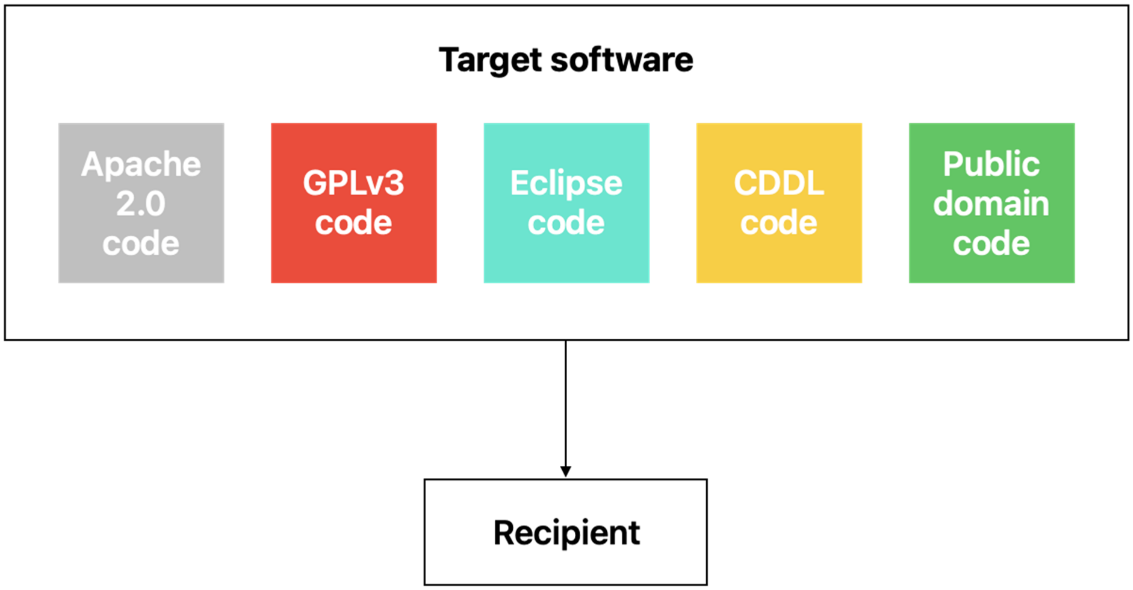


Figure 2. Example of licence incompatibility

In contrast, software provided under weak copyleft licences could typically coexist in the same program. Permissive licences place no restrictions on other code, so their use is always safe, so long as they are compatible with any copyleft licences that may be in the same code base.

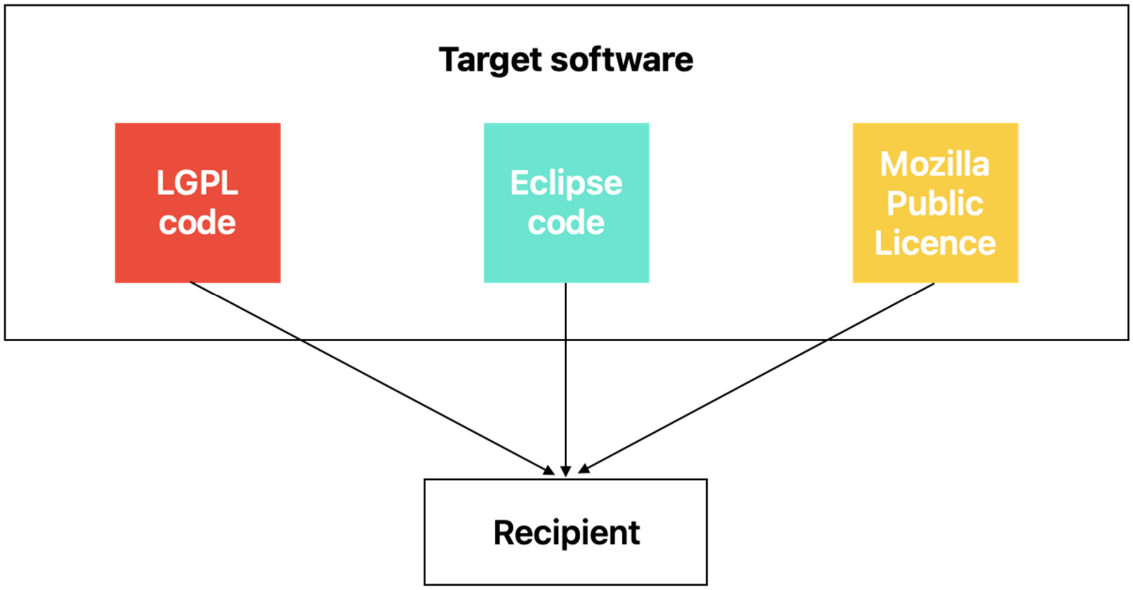


Figure 3. Example of licence compatibility

In this case each component is governed by its own licence and the code base, as a whole, does not have a single licence.

2.2.2.4. Legal implications of hybrid assets

The compatibility of licences that apply to different subject matter, ie, software, hardware and data, is not always clear. These different types of subject matter can be integrated into a single product and/or service, so ensuring compatibility of the licence conditions is crucial.

Unlike open source software licences, which rely on copyright, there is no single right applicable to hardware or data through which to channel licensing. Licence conditions applicable to these types of subject matter, when combined with software, may potentially lead to incompatibilities between the licence governing the software and these other licences.

The following practical situations illustrate some of the tensions between the licensing regimes applicable to software, hardware and data in the two most likely combinations of (1) software and data (machine learning) and (2) software and hardware (embedded systems).

Software and Data

Combinations of software and data are common but until recently data they had hardly been a subject of licensing discussions. This has changed with big data analytics which increased the incentives for companies to monetise data or for non-profit organisations to avoid its privatisation. Inspired by the success of open source software, the drafters of the first data licences have tried to apply the fundamental principles enshrined in the Open Source Definition to data. However, as discussed in D1.1,[23] data is very different from software because it is not copyright-eligible subject matter. That being said, data licences exist and certain datasets are being released under these licences, so the question is raised if developers should be concerned about potential compatibility issues between open source software licences and data licences.

This discussion has become even more relevant with the deployment of large language models in a growing number of downstream, user-facing applications. Publicly available machine learning models are now being released on a daily basis, some of them under the Apache 2.0 Licence, while others under customised licences.

Let’s take the example of Meta’s LLaMA 2 language model released under the Llama 2 Community Licence Agreement. This is a non-exclusive, worldwide, non-

[23] Ivo Emanuilov and others (n 7).

transferable and royalty-free limited licence under Meta's intellectual property or other rights owned by Meta embodied in the Llama Materials to use, reproduce, distribute, copy, create derivative works of, and make modifications to the Llama Materials.[24] The Llama Materials means, collectively, Meta's proprietary Llama 2 and Documentation (and any portion thereof) made available under the agreement. The scope of 'Llama 2' is defined as including the following:

- Foundational large language models
- Software and algorithms, including machine-learning model code, trained model weights, inference-enabling code, training-enabling code, fine-tuning enabling code and other elements of the foregoing distributed by Meta at ai.meta.com/resources/models-and-libraries/llama-downloads/

From the scope provision, it can be discerned that the licensed subject matter covers the model and the model weights but not the training or testing data. Meta recognises that the licence grant may cover not just copyright but potentially other IP rights, so the grant of rights covers intellectual property or other rights owned by Meta and embodied in the Llama Materials. The licence grant is therefore broad on purpose so as to cover subject matter that extends beyond the copyright in software.

Additional commercial terms are imposed by Section 2 of the Llama 2 Community Licence Agreement, as follows:

> *"If, on the Llama 2 version release date, the monthly active users of the products or services made available by or for Licensee, or Licensee's affiliates, is greater than 700 million monthly active users in the preceding calendar month, you must request a license from Meta, which Meta may grant to you in its sole discretion, and you are not authorized to exercise any of the rights under this Agreement unless or until Meta otherwise expressly grants you such rights."*[25]

Even though Meta consistently refers to the release of Llama 2 as 'open source', the Llama 2 Community Licence Agreement is manifestly not an open source licence because it imposes additional restrictions and limitations in contradiction to the Open Source Definition.

Custom machine learning licences can co-exist with permissive open source software licences and permissive open data licences, so long as notice and attribution requirements are met. In fact, custom machine learning licences will often attempt to

[24] Section 1.a, Llama 2 Community Licence Agreement.
[25] Section 2, Llama 2 Community Licence Agreement.

capture the full spectrum of subject matter under the heading of 'intellectual property rights or other rights'.

Reciprocal licences also do not seem to pose additional risks. One possible area of concerns is Section 1 of GPLv3 which provides that Corresponding Source includes the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by ***intimate data communication*** or control flow between those subprograms and other parts of the work (emphasis added). The concept of 'intimacy' is one of the controversial points in the licence, and the FSF has attempted to shed some light on it in the FAQ to GPLv2:

> *"(...) What constitutes combining two parts into one program? (...) We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).*
>
> *If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program.*
>
> *By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.'*[26]

The 'intimacy' between the GPL licensed code and other modules seems to refer to the question of how much these other modules depend on the GPL code. The heavier the dependence, the more likely a finding of 'intimate data communication'. At any rate, this does not seem to have a bearing on the mere aggregation of code and data in one package. The licence only extends to programs, so combining a program with data in the context of machine learning should not impinge on the licensor's rights and obligations in the software.

The following diagram illustrates the code and data landscape from an EU perspective:

[26] Free Software Foundation, 'Frequently Asked Questions about the GNU GPL v2.0' (*GNU Project*) <https://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.en.html> accessed 29 September 2023.

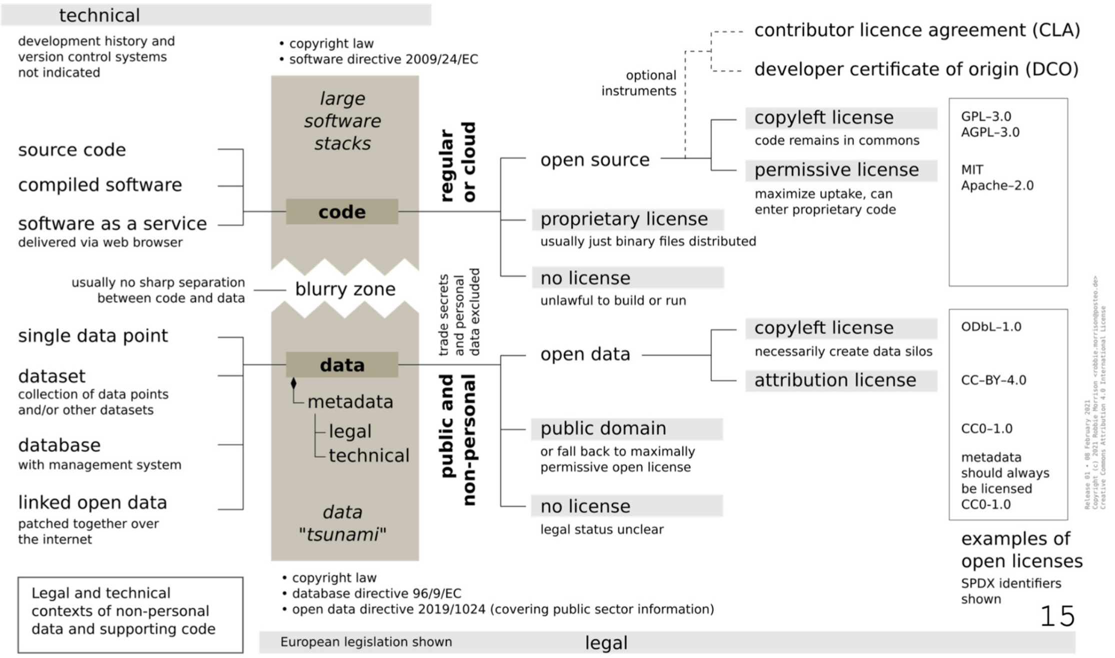

technical
development history and version control systems not indicated
• copyright law
• software directive 2009/24/EC
large software stacks
source code
compiled software
software as a service
delivered via web browser
code
regular or cloud
open source
optional instruments
contributor licence agreement (CLA)
developer certificate of origin (DCO)
copyleft license
code remains in commons
permissive license
maximize uptake, can enter proprietary code
GPL–3.0
AGPL–3.0
MIT
Apache–2.0
proprietary license
usually just binary files distributed
no license
unlawful to build or run
usually no sharp separation between code and data
blurry zone
trade secrets and personal data excluded
single data point
dataset
collection of data points and/or other datasets
database
with management system
linked open data
patched together over the internet
data
metadata
legal
technical
data "tsunami"
public and non-personal
open data
copyleft license
necessarily create data silos
attribution license
ODbL–1.0
CC–BY–4.0
CC0–1.0
metadata should always be licensed CC0-1.0
public domain
or fall back to maximally permissive open license
no license
legal status unclear
examples of open licenses
SPDX identifiers shown
15
Release 01 • 08 February 2021
Copyright (c) 2021 Robbie Morrison <robbie.morrison@posteo.de>
Creative Commons Attribution 4.0 International License
Legal and technical contexts of non-personal data and supporting code
• copyright law
• database directive 96/9/EC
• open data directive 2019/1024 (covering public sector information)
European legislation shown
legal

Figure 4. The blurry legal distinction between code and data[27]

Despite the blurry legal distinction between data and code, on the face of it, hybrid combinations of software and data do not seem to raise major incompatibility issues. In fact, precisely because there is no clear-cut legal distinction between data and code, one pragmatic solution that provides legal certainty is to licence such assemblages of code and data under a single licence, namely the licence applicable to the software.

The answer could differ, however, depending on whether models can be considered computer programs or not, and whether they may be classed as derivative works of their training data. Furthermore, if the training of models amounts to the creation of a derivative work of the training data but the resulting models are classed as computer programs, it is not clear what licences could be applied to them, ie, data licences or software licences. Similarly, if the model is considered software and the training data set is licensed under a copyleft licence, it is an open question whether the model be released under a copyleft software licence or not. These questions and possible interactions between data and code are purely speculative at this stage because there is not one uniform understanding as to where machine learning models belong in the system of recognised IP rights, if at all.

[27] Robbie Morrison, 'Open Source Software and Open Data: Open Licensing of Software and Data for Public Policy Analysis and for Collaborative Research — Release 03' (12 February 2021) <https://zenodo.org/record/4537157> accessed 29 September 2023.

Software and Hardware

Software and hardware combinations are commonplace in the semiconductor and embedded systems domains. It is in fact one of the successful business models for open source software where revenue is generated from savings made in building the device and the provision of maintenance and customisation services.

One important distinction here is between software, firmware and gateware. The difference is neatly explained in the FAQ section to the CERN Open Hardware Licence (OHL):

> *"(...) we use the word "firmware" sometimes for software which is permanently stored in a ROM or flash memory in a product. All our discussion about software applies fully to firmware because firmware is software. We do not use the word "firmware" when we discuss the design of Field-Programmable Gate Arrays (FPGA) or Application-Specific Integrated Circuits (ASICs) using Hardware Description Languages (HDL). For those cases, we use the word "gateware"*[28]

Gateware should therefore not be released under open source software licences. The CERN OHL FAQ also clarifies that the software component of the hardware is not considered a component of the hardware design and is therefore not covered by the licence conditions applicable to 'Available Components'.[29] This means that the software and hardware components of an open hardware project should follow independent licensing regimes.

2.2.3. Social Conduct

Social Conduct involves non-financial reasons for engaging in the use or development of the 3Os. Indeed, companies can engage with the 3Os for reasons beyond financial gains: their general priorities, ethos, and commitment to openness as an ethical and social choice play a substantial role. In this arena, considerations of openness as a form of "philanthropy" emerge, highlighting the broader societal impact of their engagement:

- Social impact of using open assets;
- Ethical considerations of using open assets

[28] Javier Serrano, 'CERN Open Hardware Licence FAQ' (*Open Hardware Repository*, 3 August 2023) <https://ohwr.org/project/cernohl/wikis/faq> accessed 29 September 2023.
[29] ibid.

The creation of ethical and social value through an enterprise can be a by-product of the development of open assets that was driven by financial reasons. However, recent trends in the field make the case that open source can also be prioritised over proprietary licensing strategies for other than business strategic reasons. For instance, Richard Stallman, founder of the Free Software Foundation, stated that: “Proprietary software is an injustice. Sharing is good, and with digital technology, sharing is easy”. Similarly, Jim Zemlin, executive director at The Linux Foundation, noted that: “We believe open source is a public good and across every industry, we have a responsibility to come together to improve and support the security of open-source software we all depend on”.[30]

It is also relevant to consider the recent transition towards sustainable entrepreneurship. Traditionally, an organisation’s success has been mostly related to its financial performance. In other words, entrepreneurship focused on generating wealth and economic growth while mostly neglecting environmental and social challenges (Sarango-Lalangui et al. 2018). In the last few decades, there have been extensive debates over the environmental and social role that businesses play, especially in terms of the potentially harmful impact of the current economy on our planet. This resulted in the rise of sustainable entrepreneurship (Muñoz & Cohen, 2018; Gast et al., 2017; Schaltegger & Wagner, 2011). With the rise of global economic development as an urgent issue affecting sustainable development, it has been suggested that entrepreneurship should not be predicated simply on the generation of wealth.

The 1986 Brundtland report defines sustainable development as “development to meet the needs of the present without compromising the ability of the future generations to meet their own needs” (United Nations, 1987). This definition is based on three pillars, namely, ecology, society, and economy, which take into account the so-called *three Ps*: people, planet, and profit. Sustainable entrepreneurship is also described as “an entrepreneurial approach to use the opportunities in a creative manner for economic advantages, society equity, environmental quality, and cultural preservation on an equal footing” (Majid and Koe 2012). According to Shepherd and Patzelt, the objective of sustainable entrepreneurship is to “preserve nature, life support, and community in the pursuit of perceived opportunities to bring into existence future products, processes, and services for gain, where the gain is broadly construed to include economic and non-economic gains to individuals, the economy, and society” (Shepherd & Patzelt, 2010).

[30] See https://www.rocket.chat/blog/open-source-quotes.

Levi et al. (2021)[31] outline well the ethical and social value of FOSS in terms of the democratisation of knowledge and resources. In their view, FOSS is close to the essence of public service: it is, thus, a good use of public money, one that promotes freedom of choice and avoids getting “locked in.” FOSS products and services optimise the return from public resources by permitting distributed opportunities for maintenance and updates to the code. The ease of using and reusing software solutions allows the pooling of efforts to create valuable cross-border services that are interoperable, and increase efficiency. It is easy and efficient to add features to FOSS, which can be freely shared with anyone, for any purpose.

For solutions to be stable and decisive, institutions need, through their choices in public procurements, to get involved in a virtuous circle that reverses the current trend and prevents further strengthening of the existing monopolies. Institutions can improve the FOSS code created by companies and private individuals, adding value and channelling resources into them, thereby contributing to making common accessible goods and services that are continuously evolving.

At the same time, institutions harness the knowledge available (the pre-existing source code) to make their structure more democratic, based on open data, open standards, open science and innovation, with fewer resources in the long term. In turn, this makes it possible to create services that can be supplied by all sorts of individuals, microenterprises, and SMEs, since they can offer services related to code accessible to them and to all. This virtuous circle can be applied to a democratic digitalization of Europe with strategies that invest in the creation of everyday essential digital infrastructure such as the one described in the Actions/Prototypes, with human digital rights as the raw core material for their design.[32]

In short, when it comes to reasons to base a business on the 3Os, it is important to acknowledge that companies can have different starting points in terms of the priorities above which, in turn, can influence or constrain their future developmental trajectories. To clarify, let us consider two different starting points for an enterprise:

[31] Simona Levi and et al, ‘Proposal for a Sovereign and Democratic Digitalisation of Europe’ (European Parliament 2021) <https://op.europa.eu/o/opportal-service/download-handler?identifier=dae77969-7812-11ec-9136-01aa75ed71a1&format=pdf&language=en&productionSystem=cellar&part=>.Levi, Simona, et al. 2021. ‘Proposal for a Sovereign and Democratic Digitalisation of Europe’.

[32] Relevant recent regulations: "Digital Markets Act"; "Digital Service Act". They incorporate the mindset reflected in Levi's paper.

- *Profits first:* Strategic decision (value creation/capture) may direct a company towards specific choices in terms of licensing strategies and taking up a role in a given market
- *Openness first*: Being open can sometimes be the very starting point of a company's business, constraining future business and legal choices

These two starting points are, of course, extremes on a continuum of potential drivers. Most real-world companies are in the middle: they have some strategic choices to make given their business goals but still, they may have some preferences (related to ethical values) as regards limitations/constraints of such choices, e.g., strategies that they want to avoid at all costs or areas in which they want to contribute. That is, the range of *actual* choices is limited by factors beyond business.

The social conduct component of the Extended SCP engaging with the 3Os takes into account the starting point of a company and its goals. For instance, in the case of "openness first", the pyramid of corporate responsibility described by Carroll (1991) can drastically change if not completely reversed (philanthropy and ethical principles might be prioritised over business and legal layers, for instance). In most cases, a company will need to consider all the variables of the framework in order to assess the best course of action.

2.3. Extended SCP: Performance

The extended SCP includes among the performance two additional elements that are not considered in the classical SCP: legal sustainability and social sustainability aspects. Thus, the extended SCP organises a company's performance according to three categories: Legal, Economic, and Social. In particular:

1. Economic Sustainability: involves shaping the organisational structure to optimise performance within the specific sector, guided by the 3Os principles;
2. Legal Sustainability: evaluates the effectiveness of different licensing strategies in facilitating economic and social conduct; and
3. Social Sustainability: assesses the impact of a company's choices on the broader society, taking into account their role as stakeholders in the 3Os ecosystem.

ZOOOM Licensing framework focuses on the elements of structure and conduct, as those parts serve as the primary parts that feed into the IP strategy, licensing strategy and licence management. Performance has a different role in the extended SCP, and can be monitored only based on the decisions a company makes in its Conduct. To aid in the development of the tools to guide companies in navigating the complexities in the business, legal and social landscape of open assets, ZOOOM licensing framework leaves out further development of the Performance part.

Indicators for economic, legal, and social sustainability are yet to be explored in the realm of the 3Os. However, we can expect future possibilities of assessing performance as in the classical SCP framework (profitability, growth, quality of products and services, technological progress, and productive and allocative efficiency) but adapted to the 3Os realm, where legal, ethical, and social concerns play a unique role.

Most notably, one of the relevant contributions to the reformulation of strategic action has come from stakeholder theory, a theory of organizational management and business ethics that considers multiple interests brought by the stakeholders with whom the firm relates, such as employees, suppliers, local communities, customers and others (Freeman, 1984). Stakeholder theory emphasizes ethics and values in the management of a firm, such as those related to corporate social responsibility, market economics, and social contract theory. The stakeholder view of strategy integrates resource- and market-based views, adding a socio-political layer. A common version of stakeholder theory seeks to define the specific stakeholders of a firm (normative stakeholder identification theory) then it examines the conditions under which managers treat these parties as stakeholders (descriptive stakeholder salience theory). Related to stakeholder theory is the notion of sustainability, articulated into economic, social, and environmental dimensions, and the related models of strategic action and accounting that draw on the triple bottom line approach (Slaper & Hall, 2011).

3. ZOOOM licensing framework

3.1. Open source software

3.1.1. Software intrinsics

Software is protected by copyright as a literary work. The international harmonisation of copyright under the Berne Convention has enabled open source licences to operate across borders and for licences like the GPL to me almost uniformly recognised as valid copyright licences.

The intangible nature of software has made possible global collaboration in open source projects. Anybody with a computer and programming skills could easily join an open source project. Coupled with the copyright protection that is granted automatically to any work that meets the criteria for originality, open source has become the most successful development and licensing model in the industry.

This has enabled not only global collaboration but also standardisation of the licensing terms. Standardisation is critical for creating legal certainty and predictability, which in turn make it easier for businesses to adopt this licensing model.

3.1.2. Business aspects

In a study by West (2007), it was discovered that business buyers, a key customer segment, enjoyed cost savings and the avoidance of vendor lock-in as a central value proposition. Customers had expectations of a more comprehensive "whole product" solution compared to what was provided by the open-source software (OSS) project community alone. This encompassed integration, customization, support, and more. Consequently, vendors combined both priced and unpriced complementary assets to create value.

Haff (2021)[33] offers a comprehensive analysis that delves into not only the historical evolution of the FOSS movement but also the core elements of the FOSS business model, legal considerations, and the motivations driving the movement today. The author sheds light on contemporary challenges faced by the FOSS movement. According to Haff, FOSS carries insights that extend beyond software development, touching on aspects related to how companies interact and organise themselves, as well as the interactions among individuals in their professional and personal realms.

There are numerous compelling reasons for a company to initiate an open-source project. Such endeavours can accelerate innovation, hasten time to market, facilitate the collection of new ideas, foster interoperability, recruit talented developers, and gather diverse viewpoints and contributions to enhance both code quality and product offerings. Most companies that embrace open source recognize its inherent business value and the advantages it brings in terms of efficiency, flexibility, interoperability, and the pace of innovation. By adopting open-source software, companies avoid reinventing the wheel, saving valuable time, resources, and effort while extracting more innovation from their investments.

[33] Gordon Haff, *How Open Source Ate Software: Understand the Open Source Movement and So Much More* (Springer 2021).

Furthermore, several reasons to engage with FOSS are associated with ecosystem-related benefits:

- *Innovation through collective invention*: FOSS promotes innovation by encouraging free information exchange, potentially leading to greater output and profits.
- *Economic advantage of collective effort*: FOSS projects offer significant benefits for individual participants. They share the design cost burden but reap the value of the entire project, including additions and improvements contributed by others. Additionally, collaborative projects expand the range of viable innovation opportunities for free innovators, as project costs are no longer limited to what a single individual can afford.
- *Modularity of collaboration:* FOSS collaboration thrives on modularity, where changes in one component don't disrupt other components, unlike tightly coupled organisations. This modularity aligns with the observations that communities form more easily around open-source software that follows a modular structure. Modular organisation within ecosystems enables decentralisation and opens up new avenues for conducting business.

These insights underscore the multifaceted benefits and opportunities associated with open-source initiatives, extending beyond software development and into the realms of innovation, economics, and organisational structures.

Watson et al. (2008) distinguish between five distinctive models of software production and distribution, each representing a different approach along the continuum between closed and open systems:

- *Proprietary Model:* This model has historically been the dominant force in the software marketplace. In this approach, software firms employ their own programmers to develop proprietary software, which customers then purchase. The software code is considered a valuable intellectual asset, and traditional software companies safeguard it through both physical and legal barriers, keeping it isolated from the outside world.

- *Open Community Model*: In this model, software development and support are primarily carried out by volunteers who often have limited or no commercial interests. It stands as a cornerstone of the Open Source Software (OSS) movement, characterised by a multitude of projects driven by community contributors.

- *Corporate Distribution*: While the open community model produces high-quality software, some entrepreneurs recognize that identifying suitable products, engaging with open communities for support, and acquiring the necessary support skills can be challenging for many potential OSS customers. Consequently, companies like RedHat, SpikeSource, and OpenOSX have emerged to create value and generate revenue. They achieve this by identifying the best-of-breed OSS projects, enhancing distribution methods for these products, and offering complementary services to make these OSS products more accessible to a broader market.

- *Sponsored Open Source:* In this model, corporations and foundations provide sponsorship for specific OSS projects. For instance, the Apache Software Foundation plays a pivotal role in nurturing the development of the Apache server and over 50 other OSS projects. Some corporate sponsors directly allocate development resources to OSS projects. IBM serves as a notable example, contributing developers to the Apache Web server. In certain instances, sponsored OSS projects have their roots in corporations releasing previously closed-source code and encouraging their employees to continue working on the project in its now open form. An example is Eclipse, an integrated software development environment, initially released as OSS by IBM, with IBM developers continuing to be primary contributors.

- *Second-Generation Open Source (OSSg2):* Firms embracing this model represent a hybrid between corporate distribution and sponsored OSS. These OSSg2 companies typically generate the bulk of their revenues by offering complementary services around their products, rather than selling licences for their products. Additionally, they tend to own or exercise tight control over the software code. OSSg2 models are characterised by three main features: accountability, talent-based approaches, and an emphasis on building a supportive ecosystem.

The following can be identified as business challenges for the future FOSS:

- *Cloud computing:* Centralised computing has a lot of advantages. The computers are in a controlled environment, benefit from economies of scale, and can be more easily managed. There's a reason the industry has generally moved away from server closets to data centres. However, cloud computing is also a challenge for some companies who feel that cloud service providers put them at a disadvantage. This is evident from the decisions of Elastic, MongoDB, Cockroach Labs, HashiCorp and others to switch from open source licences to source-available licences that impose restrictions on the use of their software when provided as a service.

- *Ecosystem matters:* Rise of coopetition. We see markets demanding interoperability and standards. We see more specialisation and disaggregation. However, the viability of software as a stand-alone business seems to be declining. Users increasingly don't want to explicitly pay out of pocket for software. They expect it as part of a bundle whether that means hardware or paying implicitly through advertising and other forms of monetization, which makes their attention something of a product to be sold. Software enables organisations to extract value from other things that they sell. Therefore, trademarks may be useful as part of a legal toolkit for supporting business models based on open-source development.

3.1.3. Legal aspects

This Subsection assesses legal aspects relating to IP strategy, licensing strategy and licence management. Legal implications of hybrid assets is a collection of aspects expressed with regard to FOSS under this Subsection 3.1.3, with regard to OH under Subsection 3.2.3, and with regard to OD under Subsection 3.3.3. These legal implications are further developed under Section 4, in which we take AI as an example of a hybrid technology.

An IP strategy addresses complementarities that exist between FOSS and patents, and between FOSS and trade marks. These contain aspects on mixed portfolios, implied and express patent licences in FOSS licences and making business decisions on the matter. Licensing strategy analyses FOSS business models and FOSS licensing terms to observe the dynamics that exist in the open source businesses. Licence management focuses on licence compatibility from several perspectives, eg, licence categories, compatibility tools and licence proliferation.

3.1.3.1. IP strategy: complementarities between open source software, patents and trademarks

Open source software and patents[34]

Most open source companies are also patent holding companies. Examples include not only behemoths like IBM, Microsoft, and GitHub, but also smaller companies and even start-ups. The combination of open source and patents in a single IP portfolio is therefore not uncommon in practice.

There are two perspectives on the interaction between patents and open source.[35] The first focuses on the impact of companies that own patents but are not engaged in open source licensing. These companies are often portrayed as hostile to open source. The second perspective focuses on the impact of open source licences on the patents of companies engaged in open source licensing.

Patents are often described as non-enabling rights in that the only useful right of a patent owner is to exclude everyone else from practising the invention. Patents are seen as an ineffective way of making public disclosure of useful technical information, and generally as a bad way of teaching innovation.[36] Regardless of any moral views on patent policy, patents are still considered an important part of any company's portfolio and can play a vital role in its IP strategy.

It is important to understand that patent infringement is an inbuilt risk for all software, not just open source software. The power of exclusion granted by patents is unfettered and this means that software can be infringing regardless of any acts of independent invention. Furthermore, any distribution of code that implements a patented invention exposes the distributor to patent infringement lawsuits, regardless of whether the software is distributed under an open source or a proprietary licence.[37]

Contrary to popular belief, source code secrecy is not a shield against patent infringement actions, and open source does not increase the likelihood of such actions being launched. In fact, the availability of source code makes it easier to discover if a piece of open source licensed code potentially impinges on patents. This makes it easier for plaintiffs to decide whether any claims of patent infringement

[34] The following section is based on *Patents & Open Source: Working with 'Mixed' IP Strategies* (n 12); Meeker (n 13), Chapter 13 and 14.
[35] Meeker (n 13) 189–190.
[36] ibid 192.
[37] ibid 193.

have merit or not.[38] That being said, since patents grant protection over functionality and not form, distribution in executable (binary) form in hope that source code secrecy could shield the distributor from infringement claims is a debatable strategy at best, especially in light of the breadth of claims in some software-related patents.[39] It is therefore fair to say that the availability of source code does not increase the likelihood of potential infringement actions.

Practically speaking, the main question is how companies could have patents and engage open source licensing in a compatible way. Open source licensing is not a bar to patent protection. Typically, if a company distributes open source code that implements an invention, all that happens is that the recipient also gets a licence to practise the invention.

Mixed portfolios of open source and patents - models

Van Lindberg suggests that there are three key questions when managing a mixed portfolio of open source software and patent assets:[40]

- What is the business purpose for patent protection?
- What is the scope of the open source software licence?
- How can open source software and patents be used cooperatively?

A company may want to obtain a patent for one or more of the following reasons::

- **Insurance**. The patent gives the company leverage against competitor licensing requests or lawsuits, eg, to obtain a no-charge cross licence (others-directed model). The focus is directed to others and their valuation as opposed to the patent itself. Van Lindberg remarks that the most effective patents for both insurance and assertion models are those that read on or implicate activities pursued by competitors or asserters, not on what the patent owner is doing. These externally directed patent portfolios are considered most profitable.
- **Investment**. The company develops internal R&D with the goal of improving its external valuation (internal or finance-directed model). This model is focused on getting value in terms of people as opposed to monetisation of the patent itself.
- **Assertion**. The company wants to use the patent as leverage against one or more competitors. (others-directed model). This is also an externally oriented

[38] ibid 198..

[39] ibid 199..

[40] *Patents & Open Source: Working with 'Mixed' IP Strategies* (n 12).

model where the focus lies on the most effective patents that implicate the activities of competitors.

- **Protection**. The company actually seeks to prevent use of proprietary technologies. This model is considered one of the less common, but it is also one of the models that are considered incompatible with open source. Simply put, if the proprietary technology is a key source of revenue to the business, then it should not be released under an open source licence.
- **Revenue**. The company wants to get an ongoing stream of licence royalties. This is a model driven by similar considerations to the protection model and the same advice should be followed here.

Patents are essentially a means for companies to maintain a proprietary advantage, become a source of revenue, or serve as an insurance policy. When compared with these purposes, it becomes obvious that businesses pursue very different objectives when engaging with open source software. Van Lindberg highlights five such business purposes:

- Develop a platform for innovation
- Reduce long-term risk
- Gain mindshare and/or market share
- Recruit, retain and develop people
- Reduce the cost of product/service delivery

On closer inspection, there is hardly any overlap between the purposes for getting into open source development and the purpose for developing a patent portfolio. They are simply different parts of an IP portfolio and should not be treated in a similar manner, not least because of the very different intellectual property rights that subsist in software and inventions.

Implied and express patent licences in open source licences

Many open source licences have inbuilt patent licensing provisions that, unlike traditional patent licences, are intentionally broad.[41] There are two types of patent licences: implied and express.

Implied patent licences are based on the idea that it is not fair for a patent holder to grant a copyright licence and then sue the licensee for engaging in the licensed activities covered by a patent in the patent holder’s portfolio.[42] Naturally, companies

[41] Meeker (n 13) 201.
[42] ibid.

that own patent portfolios try to avoid implied licences. For example, they may decide to grant a separate patent licence, independent of the open source licence. This cannot be done in the open source licence because of the prohibition of restrictions or reservation of rights by the Open Source Definition. The practice is likely acceptable, however, if the patent licence is granted separately.[43]

Express patent licences follow the basic principles of patent licensing, which include:[44]

- Definition of licensed patents (capture) - eg, patent owner, time period, list of patents, geographical limitations
- Definition of licensed products (object of the grant)
- Field or territory definitions (limitations in scope)

The first question in the analysis of a patent licence concerns the breadth of the patent grant. For example, a broad grant, such as that of GPLv3, requires distributors of modified versions to grant an explicit patent licence for the “whole program”. The Apache Licence 2.0 offers a narrower grant that requires a patent licence for the changes made by a distributor. If distributed source code implements a patented invention, the recipient of the distributed code may automatically receive a limited licence to the patented invention. Then there are some patent grants with ambiguous scope, such as that in the BSD licence which provides a grant to use, redistribute (sell/import), compile and modify (make).

Patent grants in open source licences are limited to necessary claims only, that is, patent claims which are essential for engaging in the activity and do not generally capture more than that. It is very important to understand where the licence grant stops, ie, to make sure that it is determined by the copyright licence. Furthermore, it must be clarified whether the patent licence is granted to the contribution only or to the entire project.

Similarly, patent ownership in most licences only extends to the patents of one entity and not affiliates or parent companies. This is critical in acquisition deals because an expansive capture provision could encumber the entire patent portfolio of the buyer when it acquires a company that has contributed to open source projects.[45] In some cases, a capture provision could include not only the patents owned by the licensor but also those that are licensable by the company, eg, patents for which the grantor has only been granted the right to sublicense.[46]

[43] ibid 202–203.
[44] ibid 203.
[45] ibid 204.
[46] ibid.

Patent grants in open source licences have one field limitation only, which is exercising the patent rights only in relation to the copyright licence in the software. Any other limitations of the field, such as territorial, commercial, or technological, would be incompatible with the Open Source Definition and are never included as a matter of principle.[47]

The time capture of patent grants in open source licences is infinite and forward-looking. This means that it captures not only the patents owned at the grant date, but also any other patents which may be prosecuted at a later date. Again, this could be a concern in acquisition deals because the buyer may end up with an unintentionally encumbered portfolio upon concluding the deal.[48]

Typically, patent provisions in open source licences include a combination of a patent licence and a defensive termination provision.[49] One typical example is Section 3 of the Apache 2.0 licence:

- The patent licence grant in Apache 2.0 captures only patents that are necessarily infringed by a contribution. Importantly, users and redistributors of the code do not grant any patent rights when they use the software or when they redistribute it without their own modifications.[50] One caveat is that the patent grant does not extend to downstream modifications and only upstream licensors can grant rights.[51]
- Defensive termination is triggered by a defensive patent counterclaim. This means that if a company exercises the open source licence and brings a claim accusing the software licensed under this licence of patent infringement, then the company would lose any granted patent licences. Importantly, however, under Apache 2.0 the copyright licence remains intact, unlike, for example, the Mozilla Public Licence.

As Van Lindberg notes, in practice every single open source software licence uses patent-related language. Verbs such as ‘make’, ‘use’, ‘offer to sell’, ‘import’, ‘export’ are typical of patent licences, not copyright licences. When a licence uses the patent verbs but does not expressly mention it grants a patent licence, there may still be express permissions to engage in patent-related activities. This is why companies try to avoid implied patent licences; they prefer to either use licences with express patent grants or grant patent licences separately.

[47] ibid 205.
[48] ibid.
[49] For example, see Section 3 of the Apache 2.0 licence.
[50] Meeker (n 13) 206.
[51] ibid.

Business decisions about patents and open source

Van Lindberg suggests that companies should rely on the 80/20 rule of business value when deciding what to patent and what to release as open source. Applied to this context, the rule states that the majority (80%) is necessary functionality, and it is not the reason why customers pay. These 80% of the functionality only support the differentiating portion and can be used as part of a cooperative strategy. It is the minority of 20% of functionality that differentiates the business offering. It is also why customers decided to buy a product. Any proprietary strategy should target this differentiating functionality. Anything below these 20% can be released as open source, for example, to reduce operational costs.

An illustrative example is the experience with OpenStack, which is a free, open standard cloud computing platform. The OpenStack code is released under the Apache 2.0 Licence, so when a company licences it in, it gives other participants in the project a licence for its contributions. In the words of Van Lindberg, companies essentially get freedom to operate with respect to their OpenStack activities, and effectively create a patent pool of sorts. In fact, open source software could be said to create a free trade zone in terms of intellectual property because of a licence that is sufficient to engage in this type of activity. As Lindberg points out, the gains from trade are the economic driver for open source.

Patent strategy is an important part of the general IP strategy, even for companies that do not intend to file patent applications. The recent lawsuit launched by the non-practising Rothschild Patent Imaging against GNOME Foundation for its Shotwell program showed that no open source project is immune from patent litigation. Van Lindberg identifies two main reasons for patent holders to assert their patent against open source projects: either because the project has cloned an important proprietary or FRAND-licensed technology, or because the project is used to make profit.

Van Lindberg draws attention to the fact that dealing with patents is mostly a community risk management issue and not that much a legal problem, and suggests the following risk-mitigation measures that should be part of any IP strategy:[52]

- Companies should be aware of and tread carefully in heavily patented areas, eg, hardware-interfacing standards or audio-visual codecs.
- Companies are advised to join patent-pooling organisations, such as the Open Invention Network for Linux-related technologies.

[52] *Patents & Open Source: Working with 'Mixed' IP Strategies* (n 12).

- Companies should adopt a contributor licence agreement (CLA) that grants patent rights from contributors. One good example is the de facto standard Apache Contributor Licence Agreement.
- Companies should use licences that include explicit patent licences.
- Companies should encourage corporate contributors, especially those with rich patent portfolios, to join their project. This can effectively create a patent pool, which, in the words of Lindberg, is defended by the code as opposed to an agreement and is very powerful in managing patent risks because it is an inherent feature of the licences which creates a freedom-to-operate area.
- Companies are advised to structurally separate their revenue from the IP ownership. This can be done, for example, by incorporating an IP holding company and an operating company. Effectively, this shields the patent portfolio from any real or perceived open source issues.
- Companies should take all patent assertions very seriously and identify non-infringement scenarios, develop a non-infringement theory, work around, or consider removal of potentially infringing assets.
- Companies should voice their concerns if threatened and leverage the available resources of open source communities to devise a strategy for defence.

Open source software and trade marks

Trade marks are essential to the success of any business. They perform the essential function of distinguishing the products of one company from the products of its competitors and serve to guarantee the origin of goods and services to the consumer.

There are, however, inherent tensions between open source and trade marks because of their perceived competing objectives. While open source promotes unfettered sharing and modification, trade marks proprietors cannot simply allow anyone to use their mark without exerting control on the resulting product (Meeker 221).

In contrast to typical trade mark management, open source projects tend to provide written guidelines on how and when a trade mark should (not) be used. Unlike traditional trade mark policies, such policies allow the use of trade marks for activities such as the naming of user or developer groups, events or promotional merchandise. Open source trade mark policies should be much more moderate in drawing the line between control and freedom. To this end, they should consider at least the following factors (Meeker 233):

- Clear policy on whether the mark can be retained in cases where the original asset (eg, code) has been modified.
- Allowed uses of the trade mark without an express permission for user or developer activities, including organisation of events or distribution of merchandise.
- Clear policy on whether logos and other protected signs should be removed from the public source tree to avoid inadvertent misuse.

There are a number of caveats which concern the mixing of marks in open source. For example, it is considered a good practice to distinguish open source software under a brand different from that of other products of the same company, especially in dual licensing scenarios. Examples include, for instance, IntelliJ IDEA and IntelliJ Idea Community Edition in the software domain, and UltraSPARC and OpenSPARC in the hardware domain. Where a company intends to start a community project that will not remain under the company's control, the best approach is to choose a completely different trade mark in order to avoid confusion.

One reasonable concern of early open source projects may be that their project name could be appropriated by a competitor who may then register it as a trade mark. In a 2018 appeal from a decision of the UK Trade Mark Registrar,[53] a UK court addressed the question whether an open source software project can generate goodwill with developers even before it is launched.

The decision suggested that open source project leaders can rely on goodwill accrued in the project name and prevent others from using it later, eg, before registering a trade mark or launching a product. The Court found that open source projects can generate goodwill in relation to attracting the provision of software developer services to undertake coding.[54] Furthermore, the Court found that individual open source developers (particularly where they can approve updates or new builds) may be responsible for generating goodwill in the project, rather than the project leader.[55] The Court considered it plausible that there may be some form of shared goodwill between the project leader and the developers.[56]

[53] Case, O-606-18, In The Matter Of The Trade Marks Act 1994, In The Matter Of Trade Mark No 3,003,117 In The Name Of Nuanti Limited, In The Matter Of An Application For Invalidation By Google Inc, and In The Matter Of An Appeal From The Decisions Of Louise White Dated February 22 2018 (O/122/18), Phillip Johnson (the Appointed Person), September 24 2018.
[54] Case, O-606-18, Paragraph 21.
[55] Case, O-606-18, Paragraph 23.
[56] Ibid.

The decision has three broader implications that can be summarised in the following way:

- Early stage projects can rely on accrued goodwill to protect the project name and prevent others acting in bad faith from registering trade marks.
- Developers may have to assign intellectual property rights to project because goodwill may accrue with individual developers.
- Trade mark searches and IP due diligence should be conducted in any case before choosing a project name, regardless of whether the project is open source or not.

3.1.3.2. Licensing strategy: Licensing models

Based on Okoli and Nguyen (2016)[57] ZOOOM D2.1 has identified the following business models for FOSS:

1. Auxiliary Services

Auxiliary services are services that go beyond just the right to use the product.

2. Corporate Development and Distribution

Paying developers to customise software to their needs and release these customisations to the FOSS community.

3. Software as a Service (SaaS)

Paying developers to customise software to their needs and release these customisations to the FOSS community.

4. Open Core, Dual Licensing and Selling Exceptions

This is a model where a “core” version of the software is released under an FOSS licence, while a version with more features is released under a proprietary licence for a fee.

5. Membership

This is a model where an individual or organisation can become a member or supporter of an FOSS development organisation, by paying a fee.

[57] Chitu Okoli and Johannes Nguyen, ‘Business Models for Free and Open Source Software’ [2015] SSRN Electronic Journal <http://www.ssrn.com/abstract=2568185> accessed 14 March 2023.

6. Crowdfunding

   This is a model where a project is financed through usually small donations of a greater number of either individuals or organisations.

7. Advertising

   This is a model where ads are displayed as part of the software, such as during the installation process, in the user interface of the software, or the manual.

8. Update Subscription

   This is a model where, in order to receive updates, patches and bug fixes, users need to become paying subscribers.

9. Selling User Data

   This is an emerging model which is based on an analysis of users behaviour inside the product which is offered under an FOSS licence free of charge.

10. Software Certification

    This is an emerging model where software is provided under a FOSS licence and can be downloaded freely, but to use the branding of the developer, a certification fee is required.

The following Figure 5 analyses the value dimensions of FOSS business models and the coverage of concurrent FOSS licence terms. The ten chosen business models for the analysis are the eight most noteworthy existing and the two most noteworthy potential business models for FOSS identified by Okoli and Nguyen (2016)[58]. In Figure 5, these are addressed from the following dimensions:

1. Object of the business:
   a. *What part of the business is directly based on code and what part on other types of business, e.g., services or brand not directly related to code?*

      Different areas are identified in different sections, the upper part of the figure (on green background) represents the software business and the lower part (on orange background) other business.

[58] Okoli C and Nguyen J, ‘Business Models for Free and Open Source Software’ [2015] SSRN Electronic Journal <http://www.ssrn.com/abstract=2568185> accessed 14 March 2023

b. *What part of the company's activities involves the FOSS offering (FOSS value creation) and what forms the business offering (value capture) within the company?*

   FOSS value creation is represented in yellow and value capture in purple.

2. Analysis of the FOSS business models to value dimensions of software:
   How do the FOSS business models position themselves in terms of value dimensions of software?

   On the x-axis you can find the ten FOSS business models and in the y-axis the value dimensions of software. The levels of the value dimensions represent increase eg, in terms of investment, and/or value and/or intellectual input.

3. Business coverage of available FOSS related licence terms:
   How well do the current licence terms cover the object and type of business?

   Coverage is expressed as vertical arrows.

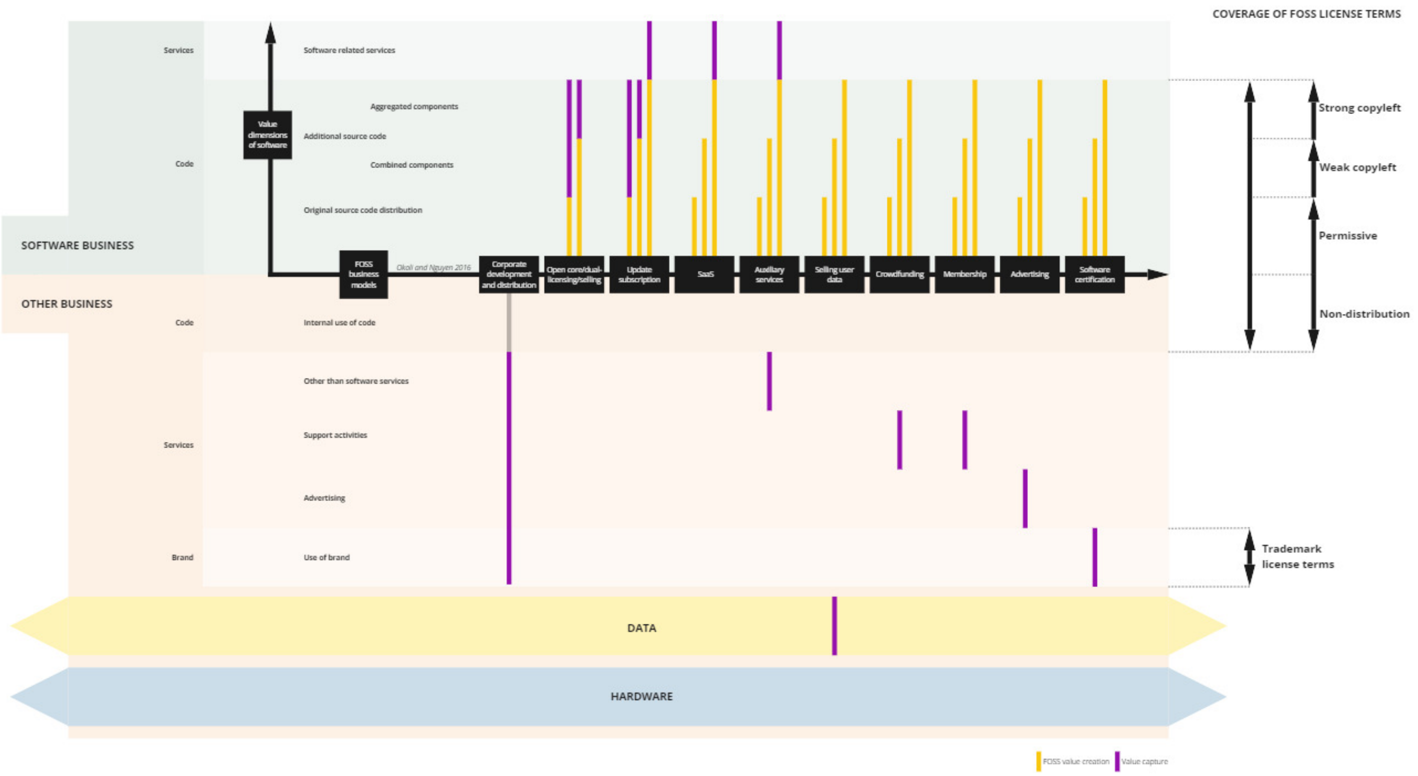


Figure 5. Analysis of value dimensions of FOSS business models and coverage of concurrent FOSS licence terms

We address similar questions from the perspective of OH and OD later in Subsections 3.2.3 and 3.3.3 to observe similarities and differences between 3Os and to be able to start addressing the needs from the perspective of hybrid open assets.

From the above visualisation, we want to highlight the core role of code in the FOSS business models and how the concurrent FOSS licence terms cover that area with diverse propositions: i.e., use that does not contain distribution, permissive licences allowing a vast amount of further business use, and weak and strong copyleft licences forming dynamics for the further use of the code within the ecosystem.

3.1.3.3. Licence management: Licence compatibility

If we imagine the obligations of each Free Software (also commonly referred to as “Open Source Software”) licence to be the demands of a dinner guest, then developing with differently licensed pieces of software can be said to be like planning an awkward dinner party.[59]

By duplicating a lot of effort, you can feed every guest present; your uncle on his low-carb diet, who wants meat and fish; your vegan sister who only eats locally grown vegetables; and your nephew who will only eat pizza. However, if your dinner guests not only limited their own diets, but also harboured a vehement disgust for everything else that they would *not* eat, it would be very difficult to bring everyone to the same table for a meal.

These difficulties arise not from the potential dinner guests being different, but rather from guests refusing to coexist with each other because of these differences. Much like difficult guests at a dinner party, Free Software (FS) licences each have their unique sets of rules and obligations. Frequently, these rules and obligations not only conflict with each other, but also exclude each other from coexisting in the same software project space. When that happens, we will have on our hands a situation of licence incompatibility.

We explore in detail what licence compatibility is, what causes situations of licence incompatibility, as well as the types of tooling that can be used to locate points of licence incompatibility in a given software project. We also briefly explore the issue of licence proliferation, and the role that licence incompatibility plays in contributing to an ecosystem where the constant creation of new Free Software licences may not necessarily be beneficial.

Free Software licence compatibility

When a software project combines two pieces of code, or merges code from one into another, it is important to pay attention to whether the terms of the licences applying to each piece of software or code allow this combination, or prohibit it. If the licences allow this combination, then the licences can be said to be “compatible” with one another. If the licences prohibit this combination, then the licences can be said to be “incompatible” with one another.

Compatible licences ensure that the code under one licence can be combined with code under another, and the resulting software can be distributed under either Free Software licence without violating the terms of the other. In other words, we say that

[59] Example adapted from Meeker (n 13) 63.

several licences are compatible with each other if it is possible to combine code under these different licences, while still complying with the terms and conditions of all these licences.

Licence compatibility is a legal framework that allows for pieces of software with different software licences to be distributed together. Consequently, incompatible licences may result in copyright infringement, as distributing these pieces of software together under incompatible licences violates the terms of one or more of the licences involved.

Licence compatibility as a result of licence categories

The reason licence compatibility arises as an issue to take note of can be boiled down to the different expectations and demands placed on users by the different categories of software licences.

On the one hand, we have proprietary (non-FS) licences, which are generally program-specific and incompatible with one another; authors of proprietary licensed programs must negotiate with one another in order to combine code. On the other hand, Free Software licences, and more specifically the reciprocal Free Software licences (copyleft) can and are prone to creating licence incompatibilities, depending on each unique situation and the licences involved. This arises primarily from the requirement in copyleft licences that the licensed code is not allowed to be used in proprietary software, and that derivatives must be licensed under the same licence terms.

Accordingly, we see that non-reciprocal Free Software licences (permissive) are generally compatible with each other, as they impose very lax or no obligations on downstream users, and as a result generally do not contain inconsistent provisions. Therefore, the copyleft licences are only compatible with other Free Software licences when one of the following conditions are met[60]:

a) The other Free Software licence (a hypothetical Licence B) does not contain any licence requirements that are not provided by the original compatible copyleft licence (a hypothetical Licence A). This is the case for example, with the 3 Clause BSD License, whereas the 4 Clause BSD

[60] Institut für Rechtsfragen der Freien und Open Source Software, 'What Is License Compatibility?' (*Institut für Rechtsfragen der Freien und Open Source Software*) <https://www.ifross.org/?q=en/what-license-compatibility> accessed 29 September 2023.

License contains an information requirement that the GNU General Public License (the GPL) and other copyleft licences do not provide.

b) The other Free Software licence (a hypothetical License C), contains a special compatibility or opening clause to allow for its distribution with code licensed under the original compatible copyleft licence (a hypothetical Licence A). This is the case for example, in section 3 of the Lesser General Public License Version 2.1 (the LGPL 2.1), which permits the use of LGPL 2.1 code under the GPL. Version 3 of the GPL contains a compatibility clause for the Affero General Public License (the AGPL), and opening clauses for the Apache License 2.0 and other licences. The European Public License (the EUPL) also contains compatibility clauses for the GPL.

Why are copyleft licences incompatible with certain FS licences?

The primary goal of the reciprocal (copyleft) licences, and in particular the GPL licences, is the promotion and furtherance of Free Software. Because of this goal, copyleft licences were crafted specifically to make it impossible to merge covered code into proprietary derivative software works.

This effect of copyleft can be seen in the two most important requirements in the GPL:

- Any derivative work from GPL covered code must itself be distributed under the GPL; and
- No additional restrictions may be placed on the redistribution of either the original work or a derivative work.

With these conditions, the GPL succeeds in spreading the four freedoms of Free Software. Once a program is covered under the GPL, these four freedoms are passed on to all other works that the code gets incorporated into, thereby making it practically impossible to use GPLed code in proprietary programs. These conditions however also mean that the GPL is incompatible with certain other Free Software licences. Additionally, many licences are not written with the intention to be GPL-compatible; historically, there have been much less transfer of files and snippets from one project to another.

Condition (a) leads to a one-way licence incompatibility for example, when trying to incorporate a component that is licensed under the GPL (reciprocal/copyleft) into a derivative work intended to be licensed under the MIT License (non-reciprocal/permissive). Because the GPL requires under condition (a) that the use of the GPL licensed component in a derivative work must result in that derivative work also being licensed under the GPL, the final licence applicable to the derivative work cannot be encompassed under the MIT License. Nevertheless, we refer to this as a one-way licence incompatibility, because a reverse of this example results in a compatible scenario: incorporating a component licensed under the MIT licence (non-reciprocal/permissive) into a derivative work intended to be licensed under the GPL (reciprocal/copyleft) is permitted, as the MIT License does not impose any conditions on how the resulting work should be licensed.

On the other hand, incompatibility under condition (b) happens when the other Free Software licence imposes a requirement that is not present in the GPL, which makes it incompatible with the GPL's condition not to add any additional restrictions on a derivative work. An example of this incompatibility can be seen in the incompatibility of Version 3 of the GPL (GPLv3) with Version 2 of the GPL (GPLv2). This is due to additional provisions in the GPLv3 that impose certain restrictions not present in the GPLv2[61].

The table[62] below illustrates the differences in obligations of reciprocal (copyleft) and non-reciprocal (permissive) licences. These differences affect directly licence compatibility.

Table 1. Licence conditions and compatibility

Non-reciprocal licences	**Scenario**	**Obligations**
	If you distribute the code	You must provide licence notice

Reciprocal licences	**Scenario**	**Obligations**
	If you distribute in binary form	You must make the corresponding source code available

[61] The main additional restrictions found in the GPL3, that are not present in the GPL2, are contained in Section 6 of the GPL3 (related to the conveyance of non-source forms of work), and Section 11 of the GPL3 (related to patent licensing).

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

If you distribute in source code form	The obligation is fulfilled
Inbound = Outbound	You must relicense the software you received on the same copyleft terms

Licence compatibility is a complex issue and in larger projects require due diligence procedures for conformity[63]. For these purposes, due diligence refers to the process of ensuring, as much as possible, that a Free Software project is complying with the terms and obligations of the Free Software licences that cover all segments of code in use. The general goal of due diligence is to ensure that, in a given Free Software project (a hypothetical Project A), the inbound rights (the rights given to Project A from incoming licensed source code) are equal to or greater than Project A's outbound rights (the rights exercised by Project A or granted to others by Project A).

In Free Software licensing, compatibility issues between the various types of licensing obligations create many diligence problems; nevertheless, the principle remains that to create software whose licensing works correctly, a given Free Software project needs to use only inbound licences that are compatible with the outbound licence. Accordingly, only outbound licences with fewer and consistent conditions should be used, as compared with the outbound licences[64].

The standardisation of licence terms makes the diligence process easier and avoids licence proliferation. The reciprocity expected in Free Software licensing determines the general principle that inbound (from contributors) and outbound (to other contributors and users) licensing terms and obligations should match. The Free Software Foundation (FSF) classifies licences on the basis of whether or not they are compatible with the GPL[65]. The chart below illustrates the terms and conditions of the four classes of licences[66]:

[63] Ibrahim Haddad, *Open Source Compliance in the Enterprise* (2nd edn, The Linux Foundation 2018) 138–142 <https://www.linuxfoundation.org/resources/publications/open-source-compliance-in-the-enterprise>.

[64] Meeker (n 13) 67.

[65] Free Software Foundation, 'Various Licenses and Comments about Them' (*GNU Project*) <https://www.gnu.org/licenses/license-list.html> accessed 29 September 2023.

[66] Chart based on David Wheeler, 'The Free-Libre / Open Source Software (FLOSS) License Slide' <https://dwheeler.com/essays/floss-license-slide.html>.

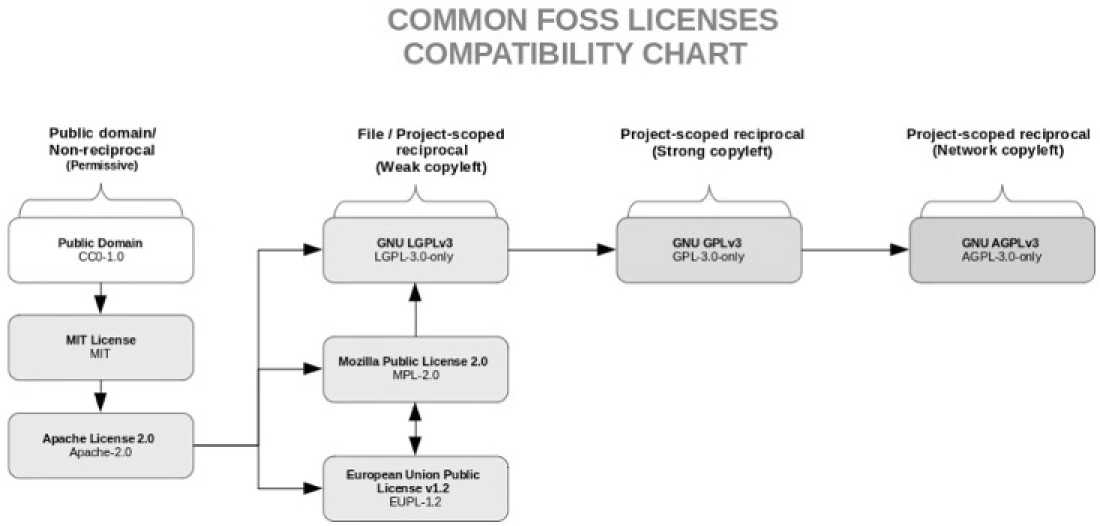

COMMON FOSS LICENSES
COMPATIBILITY CHART
Public domain/ Non-reciprocal (Permissive)
File / Project-scoped reciprocal (Weak copyleft)
Project-scoped reciprocal (Strong copyleft)
Project-scoped reciprocal (Network copyleft)
Public Domain CC0-1.0
MIT License MIT
Apache License 2.0 Apache-2.0
GNU LGPLv3 LGPL-3.0-only
Mozilla Public License 2.0 MPL-2.0
European Union Public License v1.2 EUPL-1.2
GNU GPLv3 GPL-3.0-only
GNU AGPLv3 AGPL-3.0-only

Figure 6. Common FOSS licences compatibility chart

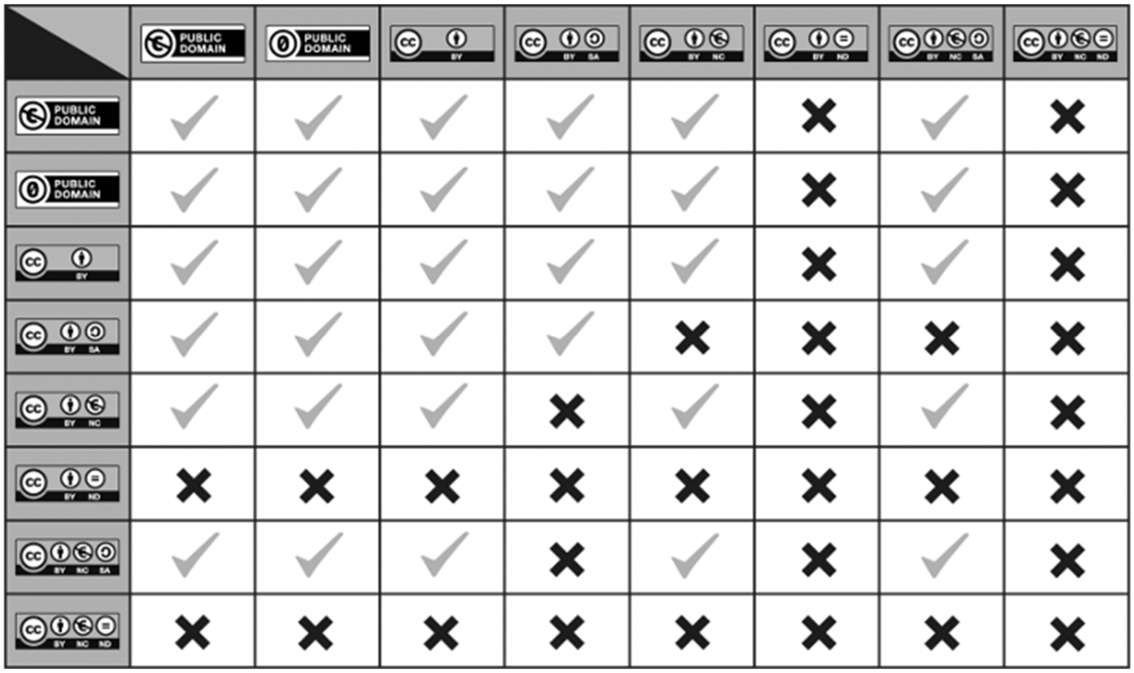
How to use this chart: To see if a software can be combined, start at their respective licences and follow to the licence boxes that can be reached by the arrows. An arrow from box A to box B means that you can combine software with these licences; the combined result effectively has the licence of B,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

Compatibility among other types of licences in software projects

In general terms, software projects may involve other types of copyrightable material beyond software code, including documentation, images, videos, music and data. The combination of licensed works, for example when making a collage or remixes of music may create incompatibility if the data licences include reciprocal (share-alike) and non-reciprocal (permissive) terms. For example, the eight commonly used Creative Commons Licences are widely used for content, but not all combinations of the licences are compatible with each other. Additionally, this is often only a one-way directional compatibility, requiring a complete work to be licensed under the most restrictive licence of the parent works. The (share-alike) licences can cause incompatibility among data and software licences as well. Both the Free Software Foundation[67] and the Creative Commons[68] have evaluation processes in place to determine the compatibility of their licences.

[67] See FSF. Licenses for Documentation. Available at: https://www.gnu.org/licenses/license-list.en.html#FreeDocumentationLicenses

[68] Creative Commons. ShareAlike compatibility process and criteria. Available at: https://wiki.creativecommons.org/wiki/ShareAlike_compatibility_process_and_criteria



	PUBLIC DOMAIN	PUBLIC DOMAIN	cc BY	cc BY SA	cc BY NC	cc BY ND	cc BY NC SA	cc BY NC ND
PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
cc BY	✓	✓	✓	✓	✓	✗	✓	✗
cc BY SA	✓	✓	✓	✓	✗	✗	✗	✗
cc BY NC	✓	✓	✓	✗	✓	✗	✓	✗
cc BY ND	✗	✗	✗	✗	✗	✗	✗	✗
cc BY NC SA	✓	✓	✓	✗	✓	✗	✓	✗
cc BY NC ND	✗	✗	✗	✗	✗	✗	✗	✗

Figure 7. Creative Commons licence compatibility chart[69]

Licence compatibility tooling

Because of the large number of Free Software licences available for use, it can often be a daunting task for any software developer to be aware of the scope of rules and obligations of all of them. It therefore is not a reasonable expectation for developers to always know the intricacies of Free Software licence terms, and to identify the compatibility of separate Free Software components. For many developers without access to immediate legal assistance, using an automated tool that tracks licence components, and that can identify and restrict incompatible licences, is often the most practical way of dealing with potential issues of licence compatibility.

To that end, the Software Package Data Exchange (SPDX)[70] serves as a valuable foundational standard. SPDX is an open standard (or format) for communicating software Bill of Materials information, which includes information related to software components, licences, and copyrights. One such way that SPDX accomplishes this is by standardising the way in which Free Software licences are generally referred

[69] Creative Commons (2023). Wiki/cc licence compatibility. Licensed under CC-BY.4.0. Available at: https://wiki.creativecommons.org/wiki/Wiki/cc_license_compatibility
[70] More information about SPDX at: *https://spdx.dev/*

to; SPDX maintains a comprehensive list of Free Software licences, which are each assigned a unique identifier.

Using the SPDX standard allows for a common format where software projects can share important data, which in turn also allows licence compatibility issues to be brought to a developer's attention automatically through machine-readable tools that can read the SPDX licence identifier. One such tool developed for licence incompatibility detection is the tool LiDetector[71], which, according to the project, automatically reads licence texts and infers rights and obligations to detect licence incompatibility in Free Software.

The European Commission also offers the Joinup Licensing Assistant (JLA) tool[72], whose primary purpose is to function as a resource allowing Free Software developers to compare and select Free Software licences based on their content. Part of the JLA's functionality also includes a compatibility checker[73], which determines how far and on which licences a particular work using or combining software components licensed under two different licences can be distributed, and if it can be distributed, under which licence(s).

In other words, the JLA Compatibility Checker's goal is to enable developers to determine the compatibility between any inbound licence (which covers third party source code that one plans to use in their project), and an outbound licence (already covering the main project source code and/or planned for distribution of the project). It accomplishes this assessment of compatibility based on many crucial classifications like obligations, permissions, prohibitions, interoperability, laws, and support. Accordingly, it aims to make finding, comparing, and selecting software licences based on their content seamless.

An e-Learning course is available online[74] for developers who wish to learn more about how to use the JLA for their licensing needs.

Additionally, the FSF also maintains an online resource on their website[75] on the compatibility of various licences with the GPL and the Free Documentation License (FDL).

[71] https://github.com/XuSihan/LiDetector
[72] More at: Joinup Licensing Assistant, Available at: https://joinup.ec.europa.eu/collection/eupl/solution/joinup-licensing-assistant
[73] More at: JLA - Compatibility Checker. Available at: https://joinup.ec.europa.eu/collection/eupl/solution/joinup-licensing-assistant/jla-compatibility-checker
[74] Available at: https://academy.europa.eu/courses/joinup-licensing-assistant-jla-elearning-course
[75] Available at: https://www.gnu.org/licenses/license-list.en.html

Licence proliferation and compatibility issues

The term "licence proliferation" refers to the phenomenon of the continued creation of new Free Software licences, adding to a large catalogue of existing licenses. Licence proliferation is generally accepted to have negative effects, and the Open Source Initiative (the OSI) has identified three main problems that proliferation has on the Free Software ecosystem[76]:

1. <u>Too many different licenses makes it difficult for licensors to choose</u>

The abundance of available existing licences makes it difficult for licensors to navigate between the various types of licences and the terms they contain, as well as to be sufficiently educated on the effects of these licences.

2. <u>Some licenses do not play well together</u>

Some Free Software licences do not interoperate well with other Free Software licences, as explained previously.

3. <u>Too many licenses makes it difficult to understand what you are agreeing to in a multi-license distribution</u>

If there are too many different individual licences covering certain distributions, that will take a lot of time for Free Software project runners to process, and adds to the difficulties in understanding the full scope of licence obligations.

Point 2 identified by the OSI highlights the problem that licence proliferation has on understanding the compatibility of Free Software licences with each other. Licence proliferation exacerbates this issue by adding new licence terms to the Free Software ecosystem, and the legal effects of such new licence terms have to be evaluated for compatibility with other existing licences. As a result, until these evaluations are completed for the new Free Software licence terms, there will be uncertainty of the effects of the new Free Software licence.

Another consideration is that newly drafted Free Software licences can often be poorly drafted[77]. More specifically, a poorly written licence will often contain licence terms that are not drafted in a manner that has taken into consideration how these

[76] Open Source Initiative, 'Report of License Proliferation Committee and Draft FAQ' (Open Source Initiative 2006) <https://opensource.org/proliferation-report/> accessed 29 September 2023.
[77] Meeker (n 13) 75.

licence terms will interact with other licences, or with the obligations and rights imposed on developers and users under the law. This means that the obligations and effects of the overwhelming majority of newly drafted Free Software licences are unknown, unclear, and can be difficult to fully understand. This in turn delays the evaluation period of the effects of the new licence, and in certain cases may even create grey areas of compatibility with no certain answers, until formally determined (for example, by statute or in a court of law).

Consequently, poorly drafted licences, licences that are not adequately stewarded or supported by a dedicated team of legal experts, and isolated and restrictive licences can create enough uncertainty such that the programmes that they cover become essentially unusable development content. They might be labelled as “free”, but the uncertainty surrounding them discourages users and developers from sharing or reusing the source code they cover, resulting in what essentially amounts to “failed sharing”[78].

Licence compatibility is an important consideration in any Free Software project that wishes to incorporate existing components or elements from other projects into and/or alongside their own original code. An existing incompatibility in your software project is a cause for concern. As we’ve explored in this chapter, licence incompatibility renders project runners incapable of fulfilling all legal obligations that they are bound to fulfil; in turn, this opens project runners up to legal liability in the form of copyright infringement, and ultimately results in the software project being non-viable in the long run.

It is therefore of vital importance that anyone looking to develop a Free Software project understands what licence compatibility is, and whether or not any inbound or outbound licensing actions undertaken by the project is a legally sound move.

3.1.4. Social aspects

Copyright, patents, trade secrets, and technological protection measures (DRM) create monopolies over knowledge, making software artificially scarce. Such limitations have an impact on open technologies and the usage of digital commons. The monopolised power can have a negative impact on fair competition and

[78] Joichi Ito, ‘The Issue of License Proliferation’ [2010] Joi Ito’s Web <https://joi.ito.com/weblog/2010/07/27/the-issue-of-li.html> accessed 29 September 2023.

consumer welfare, affecting end-user freedom of choice and individual self-determination.

Re-establishing policies centred on consumer welfare requires the promotion of distributed forms of control over technology. The growing concerns and dissatisfaction with the negative impact of such corporate power have even sparked proposals to overcome the deregulation-oriented mindset fostered by decades of neo-liberal policies, which allowed the emergence of tech oligopolies.[79] The strengthening of broader, more transparent, decentralised, inclusive, and democratic institutional arrangements over the production, development, and governance of key technologies would translate into robust policies safeguarding end-user freedom of choice, the dissolution of monopolies over device-related bottlenecks, and the promotion of interoperability policies for data and software.

Free Software, open data and open hardware seek to protect open and democratic control over assets, establishing rules over software reuse. Openness of software translates into licensing arrangements that give primacy to collective forms of sustainable and persistent access, use, and distribution of source code. With access to source code and transparent development communities, software providers can reduce development costs while remaining active participants in the development process. Additionally, end users of the software can also be active in the development process by contributing directly to upstream projects, rather than simply being passive recipients of what the software vendor delivers to them. Free Software is also considered a key element in areas related to initiatives related to open science[80], democratisation of "smart cities"[81], and alternative platform governance[82], as well in the public sector.[83]

[79] Nick Srnicek and Laurent De Sutter, *Platform Capitalism* (Polity 2017); Tim Cowen and Phillip Blond, '"TECHNOPOLY" and What to Do about It: Reform, Redress and Regulation' (ResPublica 2018) <https://www.respublica.org.uk/wp-content/uploads/2018/06/ResPublica-Report_Technopoly-and-what-to-do-about.pdf>.

[80] Frank Miedema, *Open Science: The Very Idea* (Springer Netherlands 2022) <https://link.springer.com/10.1007/978-94-024-2115-6> accessed 29 September 2023.

[81] Evgeny Morozov and Francesca Bria, 'Rethinking the Smart City: Democratizing Urban Technology' (Rosa Luxemburg Stiftung, New York Office 2018) <https://rosalux.nyc/wp-content/uploads/2021/02/RLS-NYC_smart_cities_EN.pdf> accessed 29 September 2023.

[82] Muldoon, J. (2022), *Platform socialism: How to reclaim our digital future from big tech*. London: Pluto Press.

[83] Free Software Foundation Europe (FSFE), 'Public Money, Public Code' (27 September 2023) <https://publiccode.eu/en/> accessed 27 September 2023. Katja Bego, 'A Vision for the Future Internet' (Next Generation Internet 2020) Working Paper <https://www.ngi.eu/wp-content/uploads/sites/48/2020/10/Vision-for-the-future-internet-long-version-final-1.pdf>.

In the last decade, the EU has been adjusting the behaviour of tech corporations through regulatory means, pushing forward a great quantity of legislation including the Open Internet Regulation[84], GDPR[85], DSA[86], DMA[87], and the upcoming AIA[88], and also via regulation in the fields of telecommunications and sustainability. In so doing, it is departing from a passive role based on a *laissez-faire* approach to a more active regulatory position, setting stricter behaviour rules on economic and commercial activities in digital markets.

Concepts like "device neutrality" aim to resolve the monopoly over devices, so users can have access to alternative services and content with their devices. In this sense, re-establishing end-user control over devices and fair competition in digital markets requires safeguarding software freedom in devices, protecting end-users from lock-in, and promoting end-user control over data.[89] Several aspects of this device neutrality policy concept have been incorporated in the DMA, which got translated into stricter consent rules for pre-installed apps, safeguards against vendor lock-in, and data interoperability.

3.2. Open hardware

As previously discussed, open hardware lies on a spectrum between hardwareness and softwareness. This is owing to certain specifics of hardware as an object that exists in both the digital and the physical domains.

3.2.1. Hardware intrinsics

Hardware is fundamentally different from software because it exists in material form in the sense that it is composed of atoms and not (just) bits. As discussed in D1.1,

[84] Regulation (EU) 2015/2120 of the European Parliament and of the Council of 25 November 2015.
[85] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016.
[86] Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022
[87] Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022.

[88] On December 6, 2022, the Council of the EU adopted its general approach and compromise text on the proposed Regulation Laying Down Harmonized Rules on Artificial Intelligence (the "AI Act"). More at: https://eur-lex.europa.eu/legal-content/EN/HIS/?uri=CELEX:52021PC0206. On July 14 2023 the European Parliament approved its version of the draft EU **Artificial Intelligence Act**
[89] J Krämer and R Feasey, 'Device Neutrality: Openness, Non-Discrimination and Transparency on Mobile Devices for General Internet Access' <https://cerre.eu/publications/mobile-devices-net-neutrality-internet-access/>. See also Free Software Foundation Europe, 'Digital Markets Act - FSFE' (*FSFE - Free Software Foundation Europe*) <https://fsfe.org/activities/dma/dma.html> accessed 29 September 2023.

the term ‘hardware’ could be used to refer to a lot of tangible and intangible things, from mechanical to aesthetic items.[90] This means that hardware could be covered by a potentially vast array of rights.

This diversity of subject matter, ie, from designs, through bitstreams, to physical devices, makes IP transactions with hardware a bit more challenging and less clear, at least when compared to software. For example, despite their widespread use, open source software (eg, GPL) and open culture or content licences (e.g., Creative Commons) are ill-suited for open hardware because the terminology they use does not map well to other subject matter. The concepts known from copyright law and used in many of these licences may be conceptually and practically difficult to map onto hardware.

Hardware is different from software also in terms of economics. Building a thing or reengineering existing designs in a way that does not impinge on third-party intellectual property rights is non-trivial and could be a costly exercise. At the same time, this is also what distinguishes hardware from software in that a company may have an open hardware design and still make money from selling physical devices. In the words of Andrew Katz, people are much more used to paying for atoms than for bits.[91]

3.2.2. Business aspects

Numerous studies have corroborated the positive impact of IP protection on economic growth and the emergence of new businesses (Li et al. 2021). However, it's important to recognize that hardware development is commonly perceived as more intricate compared to software development, owing to the multifaceted considerations involved, such as manufacturing, tooling, and supply chain management (Antoniou et al. 2022). This complexity prompts a critical question: how can companies effectively generate and capture value within the realm of Open Hardware (OH)?

Adding to the complexity is the evaluation of the economic impact of OH, which poses a considerable challenge. Unlike traditional economic models, OH projects often involve contributors who are not remunerated for their efforts, and individuals who create and utilize OH products might not necessarily purchase them from

[90] Ivo Emanuilov and others (n 7) 68.

[91] Based on *Open Hardware: The Next Open Revolution? | Andrew Katz | SOOCon23 Open Hardware* (Directed by OpenUK, 2023) <https://www.youtube.com/watch?v=tKcedMMe750> accessed 29 September 2023.

vendors. Furthermore, it remains difficult to quantify the number of products that have been constructed using downloaded OH designs. Consequently, there exists no official record of the value that is being generated or captured, apart from the revenues generated by OH businesses.

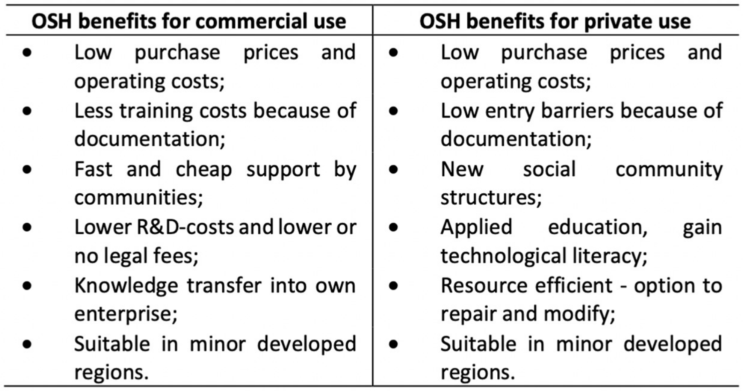
In light of these intricacies, it becomes more practical to examine the various modes of value creation rather than focusing solely on strategies for value capture (Moritz et al. 2018). This approach ensures that OH projects that contribute to value creation, even if they do not directly capture it, are not overlooked in the broader assessment of their economic significance.

Antoniou et al. (2022) identify three general characteristics of successful OSH projects: value creation, quality of output, and effective processes. More specifically, successful projects:

- create value for contributors, users, other projects and society
- generate business activity and are sustainable over time
- have a good reputation, which can be demonstrated by the ranking of projects on search engines, the number of projects, documentation, scientific paper citations, the number of views and downloads of project documentation, the number of followers/interested people, and the presence of project communities with a high level of activity (e.g., frequent participation in community forums)
- develop hardware that is highly accessible, reproducible, modifiable, performant, with transparent design, solves a problem/fulfils a need, and offers advantages over alternative products
- create high-quality documentation
- have high process openness
- follow good practices in product development and in project, community, and business management
- are transparent and committed to openness

Hildebrandt et al. (2022) summarise the main advantages of OH for commercial and private users (see the table below).

Table 2. Summary of the main benefits relating to OH. From Hildebrandt et al. (2022).



OSH benefits for commercial use	OSH benefits for private use
• Low purchase prices and operating costs; • Less training costs because of documentation; • Fast and cheap support by communities; • Lower R&D-costs and lower or no legal fees; • Knowledge transfer into own enterprise; • Suitable in minor developed regions.	• Low purchase prices and operating costs; • Low entry barriers because of documentation; • New social community structures; • Applied education, gain technological literacy; • Resource efficient - option to repair and modify; • Suitable in minor developed regions.

According to Bonvoisin et al. (2021), the revenue streams of Open Hardware (OH) projects typically originate from personal sources, external foundation grants, or crowdfunding initiatives. Interestingly, some OH projects consciously adopt a non-commercial strategy, aiming solely to sustain their ongoing activities. Conversely, other projects embrace a commercial strategy, often involving the sale of products, with the dual objective of fortifying their own endeavours and contributing to the broader open-source movement. In a more detailed analysis, Moritz et al. (2018) have delineated two fundamental modes of value creation within the OH context:

- Design Mode: In this mode, individuals create a product, openly share the design, along with all relevant information, to enable others to utilise it. Design Mode is predominantly employed in the early stages of projects, primarily because of its low financial barrier to entry. All that's required is access to a computer, an internet connection, and a degree of design proficiency.
- Production Mode: In the Production Mode, creators not only develop a product but also openly share its design. However, they take it a step further by actually manufacturing or assembling the product based on the shared design. This finished product is then offered for sale, either as a kit or ready-to-use. In this mode, value capture becomes crucial because, despite the digital representation of the physical product being freely accessible, its production demands resources such as time, money, materials, skills, and access to production facilities.

Viseur & Jullien (2022) have distinguished between two distinct strategies within the OH platform domain:

- Closed Supply-chain Platform (case study: Makerbot): In this approach, the objective is to exert comprehensive control over all components, thus allowing mastery over the platform's architectures. This strategy aligns with a quality-focused approach and leans toward a closed mode of innovation. Its primary aim is to ensure the technological consistency of the platform. This model often adopts a business model reminiscent of the "razor-blade" concept, relying significantly on the sale of raw materials. However, it demands stringent control over the overall quality of proposed solutions, including the assembly of components, as well as compatibility among hardware elements. Consequently, this approach tends to exclude the community from participating in hardware development. This strategy resembles the conventional pattern seen in the open-source sector or the broader framework of open innovation. Initially, the community may be leveraged to compensate for the company's resource limitations during the platform's launch. However, over time, the development process tends to close off as the company finds it increasingly challenging to capture the value created by the community. The firm then typically forges contractual collaborations with suppliers or invests in its proprietary technologies to ensure complete control over the innovative solutions it offers.

- Open Industry Platform (case studies: Prusa Research and Ultimaker): This strategy represents a form of innovation ambidexterity. Companies adopting this approach have developed the capability to engage in both incremental innovation and the exploration of new offerings. In the case of Ultimaker, it has even transitioned away from a fully open-hardware strategy. Meanwhile, for Prusa, an open-hardware strategy remains significant as it aids in maintaining exploratory capacities. Under this approach, the publication of machine specifications under a free licence is crucial to encourage contributions and facilitate user testing of configurations, leading to bug reporting. Notably, this strategy does not undermine value capture because the primary revenue source does not hinge on selling machines to user-developers. Instead, the main income is derived from printing solutions, for which the company possesses additional specialised human assets, such as business experts or experts in 3D technology. These assets are expensive to replicate and contribute significantly to the company's revenue streams.

3.2.3. Legal aspects

Prevailing OH licences (CERN OHL, TAPR OHL) were derived from OSS licences, but, as already explained, there is a vast difference between software and hardware. Copyright covers only the expression of an idea and not the idea itself, or its implementation in a physical product. In OH projects, this is a problem as the aim of an OH project is the implementation of an idea into a useful physical product. Even though the schematics and designs may be subject to copyright or design or semiconductor rights, the tangible product is not, and this is a major problem if distribution and commercialisation of physical devices is to be somehow controlled on the basis of exclusive rights. Even though aspects of the distribution of products are considered in some licences, it is questionable if they would be enforceable in case of infringement. Moritz et al. (2018) show that 18% of projects and companies do not use any licence at all for their products. This is critical for both makers and users. Entities that use licences (70%) rely on one of the Creative Commons licences which provide no effective protection whenever a physical artefact based on copyright-protected documentation of an idea is implemented. Only 17% use proper OH licences (CERN OHL, TAPR). These licences cover not only the digital data and documentation, but also deal with aspects regarding the physical implementation, commercial use, and distribution of the products (like patent grants). In addition, 5 companies applied for patents (for defensive purposes), which does not necessarily contradict the idea of open source when anyone is granted permission to use the patented idea (eg, through open patents non-assert pledges). Overall, 26 projects and 23 companies (63%) comply with the strict notion of open source by using licences that explicitly permit commercial use (eg, CC-BY-SA).

Licensing models in the spectrum of 'hardwareness'

As discussed in D1.1, in practice, most commercially viable open hardware projects are developed under permissive open source licences. Permissive licensing drives commercial adoption, especially in domains where deep patent portfolios are commonplace, such as semiconductors. Reciprocal licences are often perceived as creating a greater risk of unwanted patent exposure and are avoided in most commercial settings.

Simply put, permissive licensing avoids the risk of asking, for example, semiconductor companies to 'grant back' and expose their patent portfolios to unwanted risk. Community participation of big companies in the open hardware domain should be encouraged through other incentives, for instance, if they can recognise a clear commercial benefit in collaborating with others.

When looking at licensing models for open hardware, we distinguish the two main types of models, namely permissive and reciprocal. However, because hardware lies on a spectrum between softwareness and hardwareness,[92] we need to distinguish the

[92] Blind, Knut and others, 'The Impact of Open Source Software and Hardware on Technological Independence, Competitiveness and Innovation in the EU Economy' (Publications Office of the European Union 2021) Final Study Report 341 <https://ec.europa.eu/newsroom/dae/redirection/document/79021>.

licensing models based on whether the hardware is closer to the software or the hardware end of the spectrum.

It has been suggested that the development of hardware (and software) involves a continuous cycle of designing, building, testing, re-designing etc.[93] When ready, the cycle switches to productisation. While for software all of these activities occur in the digital domain, hardware involves activities like building physical prototypes where much of the testing occurs in the physical domain.[94] The degree of involvement of knowledge creators in open source development models is dependent on the level of hardwareness. As reported in a recent study commissioned by the EC, "[t]he "harder" the hardware, the more likely it is that the design and development work is ultimately carried out in the same way as it is carried out by proprietary developers by a centralised research and development operation as opposed to a community of collaborators."[95]

The following figure illustrates where some of the most popular categories of hardware lie on the hardwareness spectrum:

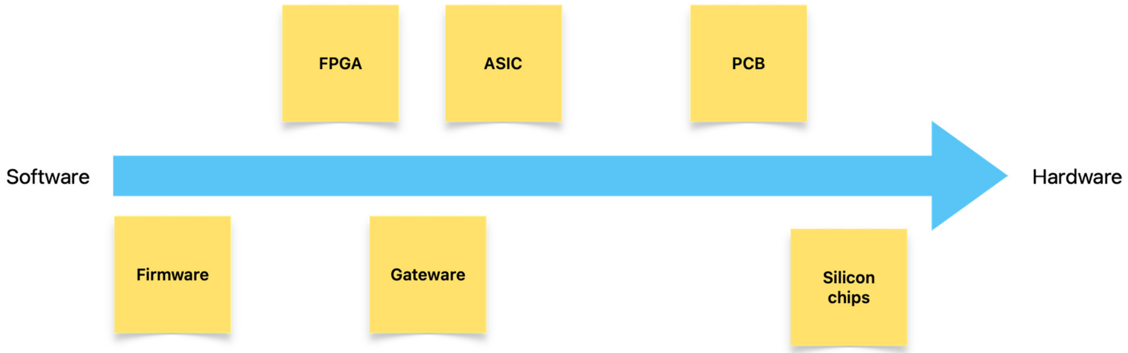


Figure 8. The hardwareness spectrum

Permissive open hardware licensing models

As explained, permissive licensing drives commercial adoption. This is particularly evident in the domain of semiconductor chips. One example is the activities of the

[93] ibid 55.
[94] ibid.
[95] ibid 61.

industry members of the OpenHW group. OpenHW group is a not-for-profit, global organisation where hardware and software designers collaborate in the development of open-source cores, related IP, tools and software. The organisation is focused exclusively on development of open source, RISC-V microprocessor cores and the associated accelerators, interfaces, enabling hardware and System on Chip (SoC) platforms, and software toolchains. In the words of Duncan Bees, director of technical programmes for the OpenHW Group:

"The output of an OpenHW processor core project is typically a set of open source Intellectual Property (IP) that includes fully verified Register Transfer Level (RTL) code, a user manual and test benches that together can be used as the heart of a semiconductor device design. OpenHW projects are developed under permissive open source licences, such as Solderpad 2.0 (which builds on Apache 2.0), that provide implementers the freedom to innovate, to customise, and to commercialise. OpenHW's use of the Eclipse Development Process (EDP) ensures that IP adopters have full confidence in the provenance of open source contributions and the integrity of project outputs."[96]

The adoption of the Eclipse Development Process[97] in the development activities is important for three reasons:

- Contributor agreements including developers' certificates of origin;
- Election of committers based on demonstrated technical and procedural merit;
- Vetting of contributions for compliance with the project's open source licences and intellectual property rules

As discussed in D1.1,[98] the Solderpad licence is perhaps the most popular permissive open hardware licence. As a licence based on Apache 2.0, it offers predictability and legal certainty for organisations and their legal counsels. In its latest version 2.1, Solderpad supplements the following definitions from the Apache 2.0 licence:

- References to 'authorship' shall read "authorship or design"
- References to 'copyright owner' shall read "Rights owner"
- References to 'copyright statement' shall read 'copyright or other statement pertaining to Rights'

[96] https://newsroom.eclipse.org/eclipse-newsletter/2022/january/openhw-group-open-approach-microprocessor-design
[97] https://www.eclipse.org/projects/dev_process/
[98] Ivo Emanuilov and others (n 7).

Solderpad Licence v2.1 adds the following new definition of ‘Rights’, which means copyright and any similar right including design right (whether registered or unregistered), rights in semiconductor topographies (mask works) and database rights (but excluding Patents and Trademarks).

Essentially, the Soldepad licence is limited to copyright, design rights, semiconductor topography rights, and database rights. Patents and trade marks are explicitly excluded.

The licence sets the terms and conditions for use, manufacture, instantiation, adaptation, reproduction, and distribution. The licence is perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable and allows the licensee to reproduce, prepare derivative works of, make, adapt, repair, publicly display, publicly perform, sublicense, and distribute the work and such derivative works in source or object form and do anything in relation to the work as if the rights did not exist. The licence is conditioned upon the licensor being the owner of the rights or entity authorised by the owner of the rights that is granting the licence.

The Solderpad Licence reflects the hardwareness spectrum discussed above. The definition of ‘object’ covers mechanical transformation or translation of a source form or the application of a source form to physical material, including but not limited to compiled object code, generated documentation, the instantiation of a hardware design or physical object or material and conversions to other media types, including intermediate forms such as bytecodes, FPGA bitstreams, moulds, artwork and semiconductor topographies (mask works). The licence defines ‘source’ as the preferred form for making modifications, including but not limited to source code, net lists, board layouts, CAD files, documentation source, and configuration files.

As an alternative to Solderpad v2.1, companies may also opt for the CERN Open Hardware Licence v2 in its permissive variant (CERN-OHL-P). CERN OHL focuses on design documentation, so the user rights under this family of licences are granted once the user performs an act that would impinge on the exclusive rights in the design documentation. CERN-OHL-P requires that the licensor keeps all notices and provides a copy of the licence. If these two conditions are met, the licensor may convey both covered and modified covered source under different licence terms, including proprietary.

From a commercial perspective, permissive open source licences like Solderpad v2.1 are perhaps the best and safest choice for open hardware projects on commercial scale for two reasons. The first is the flexibility of the terms and conditions and the possibility for customisation and commercialisation. The second is the fact that the industry is familiar with and trusts the Apache 2.0 licence.

Reciprocal open hardware licensing models

The de-facto standard reciprocal open hardware licence is the CERN-OHL v2 in its weakly and strongly reciprocal variants. The main concern with this family of reciprocal licences is the limited ability to combine components released under different licences as well as the patent and retaliation clause.

The following diagrams illustrate two examples of how the reciprocal variants of the CERN-OHL v2 apply to PCB design and FPGA/HDL/ASIC designs.

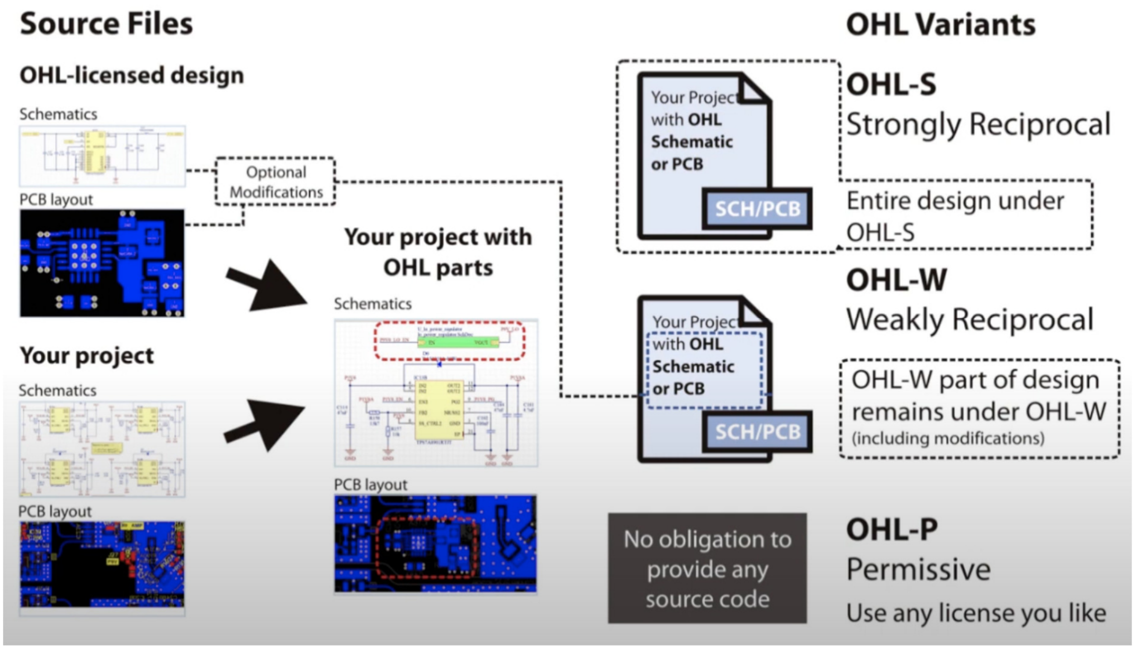


Figure 9. CERN-OHL v2 applied to PCB designs. Source: Javier Serrano, CH Open Business Event - CERN Open Hardware Licence v2, available at https://www.youtube.com/watch?v=6wvEgQ5iWoc[99]

[99] *CH Open Business Event - CERN Open Hardware Licence V2* (Directed by CH Open, 2020) <https://www.youtube.com/watch?v=6wvEgQ5iWoc> accessed 29 September 2023.

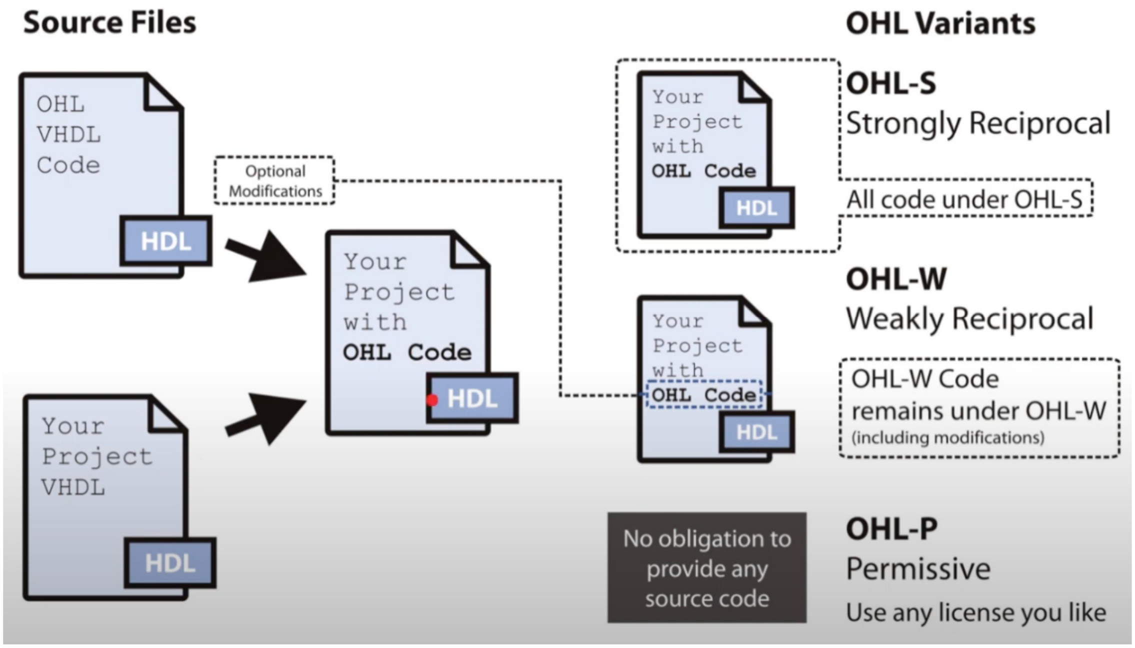


Figure 10. CERN-OHL v2 applied to HDL/FPGA/ASIC designs. Source: Javier Serrano, CH Open Business Event - CERN Open Hardware Licence v2, available at https://www.youtube.com/watch?v=6wvEgQ5iWoc[100]

The CERN-OHL family of licences allows for the use of proprietary primitives (eg, sub-components) that may be incorporated by proprietary toolchains, the only requirement being that the toolchain must be accessible, even if only for a fee.[101]

As discussed in D1.1,[102] reciprocal licences are perceived as creating business risks. Nurturing a community of commercially active companies around a reciprocal licence is therefore much more challenging compared to permissive licences. This is especially the case for hardware where the costs of customisation, verification, manufacturing and preparation of regulatory compliance dossiers in regulated industries, such as healthcare or automotive, could be prohibitively high.

The following Figure 11 (Attachment Y) analyses the value dimensions of OH business models and the coverage of concurrent OH licence terms. Two of the chosen business models for the analysis are identified by Moritz et al. (2018)[103] and

[100] ibid.

[101] *Open Source Software and Hardware: Community, Business Models and Licensing by Andrew Katz* (Directed by Instituto de Nanosistemas UNSAM -, 2022) <https://www.youtube.com/watch?v=c-TeNLTRtAg> accessed 29 September 2023.

[102] Ivo Emanuilov and others (n 7).

[103] Manuel Moritz and others, 'Value Creation in Open-Source Hardware Communities: Case Study of Open Source Ecology', *2016 Portland International Conference on Management of Engineering and*

two additional ones by Arancio & Molloy (2021)[104]. In Figure 11, these are addressed from the following dimensions:

1. Object of the business:

a. *What part of the business is directly based on design files and what part on other types of business, e.g., brand, standardization efforts or other not directly related to hardware?*
 Different areas are identified in different sections, the upper part of the figure (on blue background) represents the hardware related business and the lower part (on orange background) other business.

b. *What part of the company's activities involves the OH offering (OH value creation and what forms the business offering (Value capture) within the company?*
 OH value creation is represented in yellow and value capture in purple.

2. Analysis of the OH business models to value dimensions of hardware:
 How do the OH business models position themselves in terms of the value dimensions of hardware?
 In the x-axis you can find the ten OH business models and in the y-axis the value dimensions of hardware. The levels of the value dimensions represent increase e.g., in terms of investment, and/or value and/or intellectual input.

3. Business coverage of available OH related licence terms:
 How well do the current licence terms cover the object and type of business?
 Coverage is expressed as vertical arrows.

Technology (PICMET) (IEEE 2016) <http://ieeexplore.ieee.org/document/7806517/> accessed 17 January 2023.

[104] Julieta Arancio and Jenny Molloy, 'OPEN HARDWARE IS READY TO HELP TECHNOLOGY TRANSFER OFFICES MAXIMISE THE IMPACT OF ACADEMIC RESEARCH' (*Google Docs*, 2021) <https://drive.google.com/file/d/1E9p0_HOj9Q3QHsIr8l-NID2bYkYPxiKD/view?usp=embed_facebook> accessed 2 January 2023.

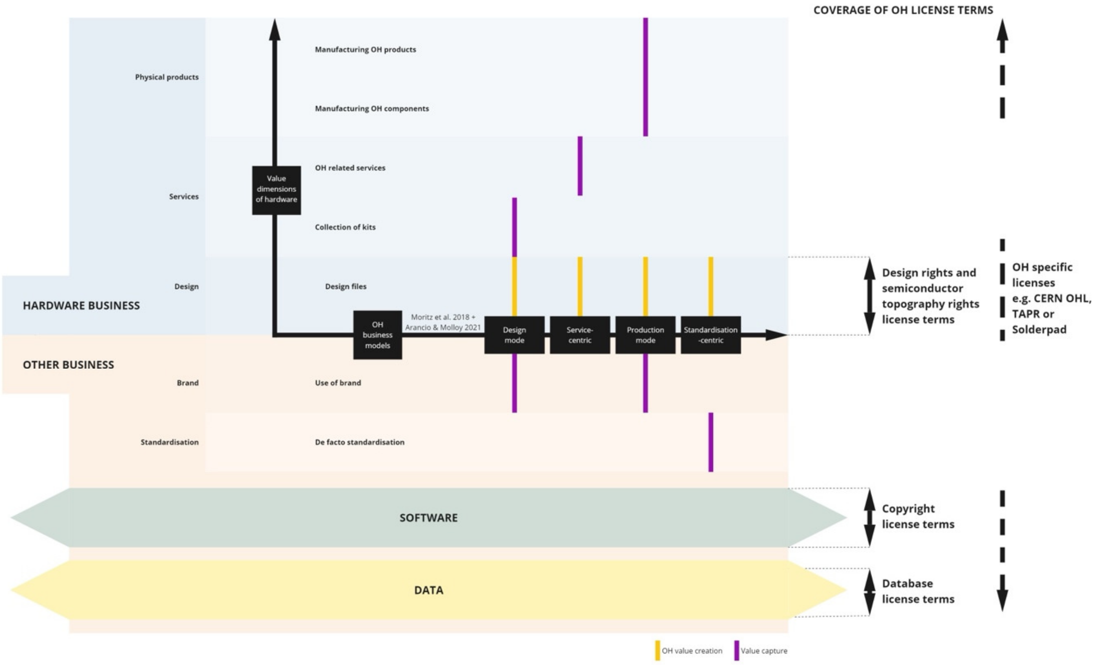


Figure 11. Analysis of the value dimensions of OH business models and coverage of concurrent OH licence terms

We have addressed similar questions from the perspective of FOSS earlier in Subsection 3.1.3 and will address them from the perspective of OD later in Subsection 3.3.3 to observe similarities and differences between 3Os and to be able to start addressing the needs from the perspective of hybrid open assets.

From the above visualisation, we want to highlight the differences compared to the one on FOSS. In OH business, there is an additional layer of manufacturing physical products. In addition, it should be noted that OH business can interlink to other business such as FOSS.

3.2.4. Social aspects

Key aspects of OH are transparency, accessibility, and replicability (see Antoniou et al. 2022; Balka et al. 2010; Bonvoisin et al. 2021; Moritz et al. 2018):

- *Transparency*: The possibility for any interested person to have unrestricted access to information sufficient for understanding the product in detail.

- *Accessibility*: Any person interested is able to actively participate in developing the product by editing design information.
- *Replicability*: the product can be physically reproduced.

Li et al. (2021) interviewed many OH firms and showed that the reasons for going open result from: a) intrinsic factors, such as entrepreneurs' sense of moral obligation, altruism and b) extrinsic motivations, such as market obligations, reduced time-to-market, lowered R&D costs and lowered customer support cost (see the figure below).

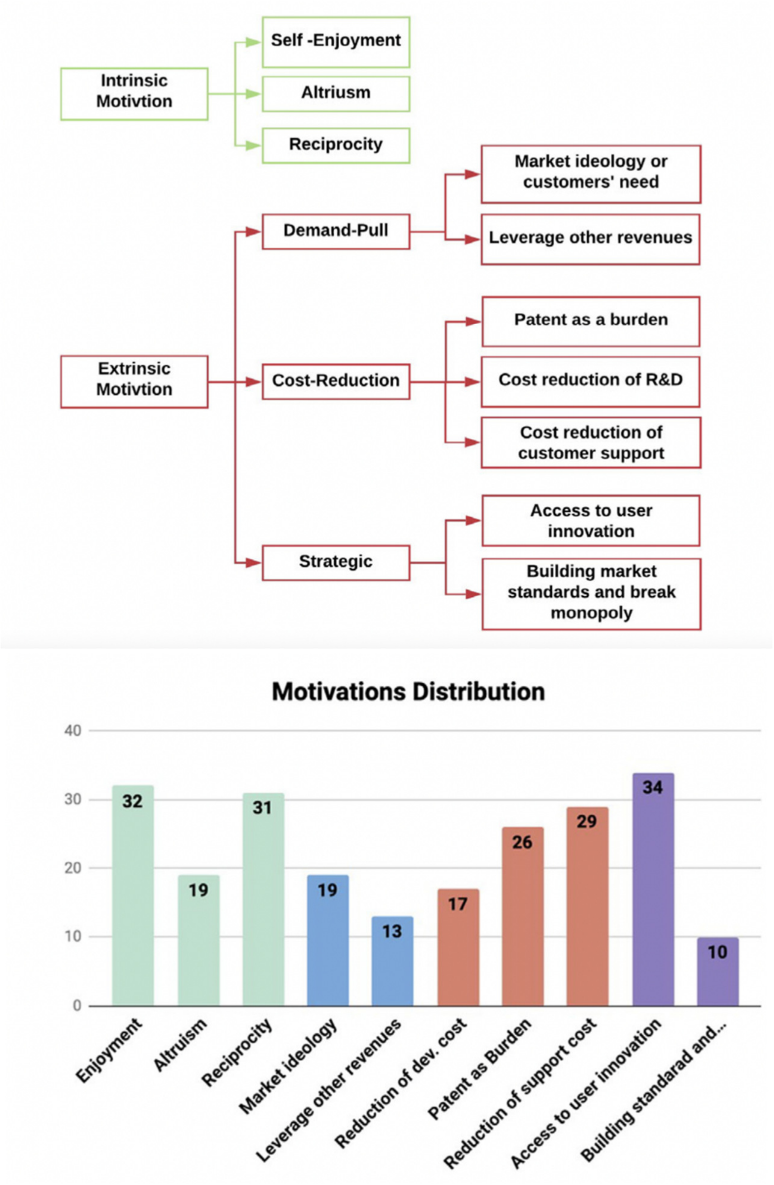


Figure 12. Intrinsic and extrinsic motivations for the involvement in OH. From Li et al. (2021).

3.3. Open data

In the ZOOOM licensing framework, data plays a crucial role, either as solely open data or as part of the hybrid forms of 3Os. An example of such hybrid forms integrating data and software is AI, which is studied in more detail under Section 4 below.

Before delving into hybridity, it is important to understand what makes data different from software or hardware. This distinction affects all aspects of the ZOOOM licensing framework, ie, economic, legal and social. Looking from an economic and social perspective, value creation from data is different compared to the value creation in software or hardware related business. Multifaceted value creation puts more emphasis on the ecosystemic aspects. Also, from the legal perspective, there are profound differences between the IP protection base of data compared to software or hardware. To name some of these differentiating factors: data, as such, does not enjoy intellectual property protection, whereas source code of software is protected by copyright; and data comes in different intangible forms – some of it even in machine-readable forms that cannot be read by humans – whereas hardware typically has a material form and is a potential candidate for patent protection. The IP basis of data affects the way business may be generated from data and how the licensing of data should take place.

In the following subsections, we will look into the economic, legal and social aspects relating to open data in more detail and analyse how well the business aspects represented by different OD business model archetypes meet the current OD licensing practices.

3.3.1. Data intrinsics

Looking at the concurrent trends regarding data sharing beyond the boundaries of a single organisation, part of which the sharing of OD is, we can observe two trends. Firstly, there is a trend from mere data sharing towards data productization and data servitization. And secondly, the role of data ecosystems is becoming more prominent. These trends can be observed for instance in the European strategy for data[105], the regulations included therein[106] and in initiatives around European data spaces[107]. The roots and trends of open data have been studied in more detail in ZOOOM D1.1.[108]

The trends towards higher value services based on data and data ecosystems bring in focus two differing approaches regarding openness of data. That becomes evident when looking into data sharing at large, ie, all activities that take place beyond the boundaries of one single organisation. This includes open data (OD) within the meaning given by the Open Knowledge Foundation (OKFN) Open Definition[109]. But it also includes data that is not strictly speaking OD, like openly accessible data, and data sharing within open data ecosystems (ODEs) – or even areas that are quite close to closed data, eg, temporary sharing of real-time data within a single value chain. The latter kinds of data sharing are in contradiction with the OKFN Open Definition, but nevertheless represents a change in the concurrent practices based on closed data that is only shared within a single organisation and even being embedded in silos within each organisation.

The above trends give us perspective to assess what makes data different from software or hardware. First, it needs to be acknowledged that data comes with a data spectrum[110], varying from closed to open and having different forms of data sharing in between. The other aspect, that makes data different from software and hardware, is its ambiguity in the forms it takes and in terms of value creation based on that. Data can vary from one-off raw data to online real-time data sharing; it can be machine-readable or expressed in human-readable format; and it can possess no or full IP protection. This ambivalence makes the value creation from data burdensome and unpredictable.

[105] COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS A European strategy for data 2020

[106] See eg, Regulation (EU) 2022/868 of the European Parliament and of the Council of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act) (Text with EEA relevance) and Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on harmonised rules on fair access to and use of data (Data Act) 2022

[107] See eg, 'Staff Working Document on Data Spaces | Shaping Europe's Digital Future' (14 February 2022) <https://digital-strategy.ec.europa.eu/en/library/staff-working-document-data-spaces> accessed 23 September 2023

[108] Ivo Emanuilov and others (n 7).

[109] Open Knowledge Foundation (n 4).

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

3.3.2. Business aspects

As open data is often pursued by public authorities, such as cities and public officials, open data has emerged as an asset to be utilised by different organisations. In some cases companies also produce open data sets back to the community. In this connection open data is defined according to the OKFN Open Definition, i.e. to be freely accessible and re-usable by any party, and in this form open data may stimulate innovation[111].

Potential value propositions related to open data are focused mainly around using the data as a resource or support for the core business. New insights generated from the open data may help businesses to:

- develop new products (value disciplines: usefulness).
- improve their supply chain related efficiencies (value disciplines: process improvement)
- improve their existing products or services (value disciplines: performance)
- improve their customer understanding and gain better market insights (value disciplines: customer loyalty)

Usefulness refers to business model archetype, which uses open data directly to create something of value for the customer. Business models like freemium, premium, subscription and dual licensing are all associated with this approach. Companies can enrich the open data, create additional services on top of open data or provide customised services based on open data.[112]

Open data can also be used for internal development and efficiency (namely process improvement and performance). These are business models that are using open data for improving existing processes and creating cost savings. Examples here include automated tasks and supply chain management, as well as improved internal decision making. Performance refers to releasing open data to support primary business objectives of the company, for example by releasing data about the main products or services of the company.[113]

[111] van der Broek, T.A., Rijken, M. and van Oort, S.H., 'Towards Open Development Data : A Review of Open Development Data from a NGO Perspective' (2012) TNO 2012 P10098 <http://resolver.tudelft.nl/uuid:c1ef3a5a-155d-4139-bb47-360a401ca339>.

[112] Adapted from Fatemeh Ahmadi Zeleti and Adegboyega Ojo, 'The 6-Values Open Data Business Model Framework' in Adegboyega Ojo and Jeremy Millard (eds), *Government 3.0 – Next Generation Government Technology Infrastructure and Services*, vol 32 (Springer International Publishing 2017) <http://link.springer.com/10.1007/978-3-319-63743-3_9> accessed 24 January 2023.

[113] Adapted from ibid.

Finally, open data can be used to improve customer understanding and loyalty. The business model focuses on using open data to improve and enhance the original value proposition of the company. Examples include using open data from social platforms to generate improvements to service processes based on customer complaints and creating personalised experiences for customers.[114]

Value creation from open data requires an ecosystem consisting of different actors in an open data ecosystem. According to Kamarioutu & Kitsios (2022) these can be grouped into three main entities[115]):

- data and infrastructure providers
- application developers
- end users.

Data and infrastructure providers are entities that give data freely available to others in the ecosystem and may deploy a specific licence on the data to restrict or limit the use of data, by using for example public licences. These are often public organisations or officials, who do not necessarily require financial compensation for the data, but rather want to see new business and use cases developed from the open data. Infrastructure providers provide tools for the open data value chain to work, for example they create marketplaces for data exchange to happen. Application developers provide applications and services that are enabled by open data. Their applications are used by end users, who can be consumers, citizens or enterprise users.[116]

3.3.3. Legal aspects

As software, hardware and data each have a clear difference in the basis of their IP rights, many legal aspects with regard to OD stem from this difference. Data as such is not protected by copyright; however, in Europe, databases are covered by the European database protection[117]; and if data manifests itself as content, it can get protection through copyrights. In addition, protection through trade secret legislation

[114] Adapted from ibid.

[115] Maria Kamariotou and Fotis Kitsios, ‘Bringing Digital Innovation Strategies and Entrepreneurship: The Business Model Canvas in Open Data Ecosystem and Startups’ (2022) 14 Future Internet 127.

[116] ibid.

[117] Consolidated text: Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases

and contractual practices affect how protected data is. One could say that best protection for data would be non-disclosure, i.e. keeping it fully closed within a single organisation. The differences in the IP rights basis is studied in more detail in ZOOOM D1.1.[118]

However, a closed approach for data contradicts the need for adding value to data. To a certain extent, this added-value can be generated in silos, such as within a single organisation, but the biggest potential in terms of value comes from novel ways to use data, which becomes possible through different forms of data sharing. And with data sharing, we can start to discuss the licensing of data and terms relating thereto.

As identified in ZOOOM D1.1,[119] there are several problems in open data licensing, e.g., the level of intellectual input is ambivalent; the traditional licences contain irrelevant terms or terms that are difficult or even impossible to adhere to; there are challenges with respect to data governance and compliance; and human-machine interaction is problematic – to name some.

In the concurrent licensing scene around OD we can distinguish between the traditional *property-focused licensing* and licensing focused more specifically on certain specific emerging technology, e.g., AI or blockchain technologies. The latter take a more detailed approach to the technological components of the emerging technology, the interactions between these components, and, ultimately, manage to leap towards the potential added value that can be achieved through data sharing and licensing. Hereinafter, we call the latter *value-focused licensing*.

Property-focused licensing addresses data sharing either solely as licensing data and/or databases or from the content-based copyright-perspective covering also data and/or databases. If we compare these to the main categories that we have for FOSS licensing, ie, permissive licences and copyleft licences, the following general comparison can be made:

1. Open data licences comparable to FOSS permissive licences:
   - Dedication to public domain (without attribution requirements): CC0-1.0 (content), PDDL-1.0 (databases)
   - With attribution requirements: CC-BY-4.0 (content), ODC-By-1.0 (databases)
2. Open data licences comparable to FOSS copyleft licences:

[118] Ivo Emanuilov and others (n 7).
[119] ibid.

- Share-Alike:
  CC-BY-SA.4.0 (content), ODbL-1.0 (databases)

It is noteworthy that as FOSS-licences focus on software code (protected by copyright), the above licences focus either on content (e.g., text or visuals protected by copyright) or on databases (protected by the database protection). Even when the above mentioned licences partly also cover data embedded in the content or the database, none of the above takes a more focused data related perspective that would allow addressing the value creation by data in more detail.

When leaning towards value-focused licensing, the focus needs to shift towards the value creation mechanisms relating to the particular field of technology and dynamics that exists within the ecosystem of actors and the community. This forms the basis, on which the value-focused licensing models need to be built.

Some of these novel type of approaches, specifically focused on AI (Montreal data licence tool[120]) and blockchain (Cryptographic autonomy license[121]) have been addressed in ZOOOM D1.1.[122] In addition this will be further elaborated, especially from the perspective of restrictions, under Section 4 below with regard to AI.

The following Figure 13 analyses the value dimensions of OD business model categories and the coverage of concurrent property-focused OD licence terms. The categories chosen for the analysis are the five open data business model categories identified by Zeleti & Ojo (2017)[123]. In Figure 13, these are addressed from the following dimensions:

1. Object of the business:
   a) *What part of the business is directly based on data and what part on other types of business, e.g. processes, products and services not related to data?* Different areas are identified in different sections, the upper part in the figure represents the data business and the lower part other business.
   b) *What part of the company's activities involve the OD offering (OD value creation) and what forms the business offering (value capture) within the*

[120] Misha Benjamin and others, 'Towards Standardization of Data Licenses: The Montreal Data License' (arXiv, 20 March 2019) <http://arxiv.org/abs/1903.12262> accessed 2 November 2022.
[121] Arthur Brock, 'Understanding the Cryptographic Autonomy License' (*Holochain*, 22 February 2019) <https://medium.com/holochain/understanding-the-cryptographic-autonomy-license-172ac920966d> accessed 26 February 2023.
[122] Ivo Emanuilov and others (n 7).
[123] Adapted from Zeleti and Ojo (n 112).

company?

OD value creation is represented in yellow and value capture in purple.

2. Analysis of the OD business model categories in terms of the value dimensions of data:
How do the OD business model categories position themselves in terms of the value dimensions of data?
In the x-axis you can find the OD business model categories and in the y-axis the value dimensions of data. The levels of the value dimensions represent increase eg, in terms of investment, and/or value and/or intellectual input.

3. Business coverage of available data related licence terms:
How well do the current licence terms cover the object and type of business?
Coverage is expressed as vertical arrows.

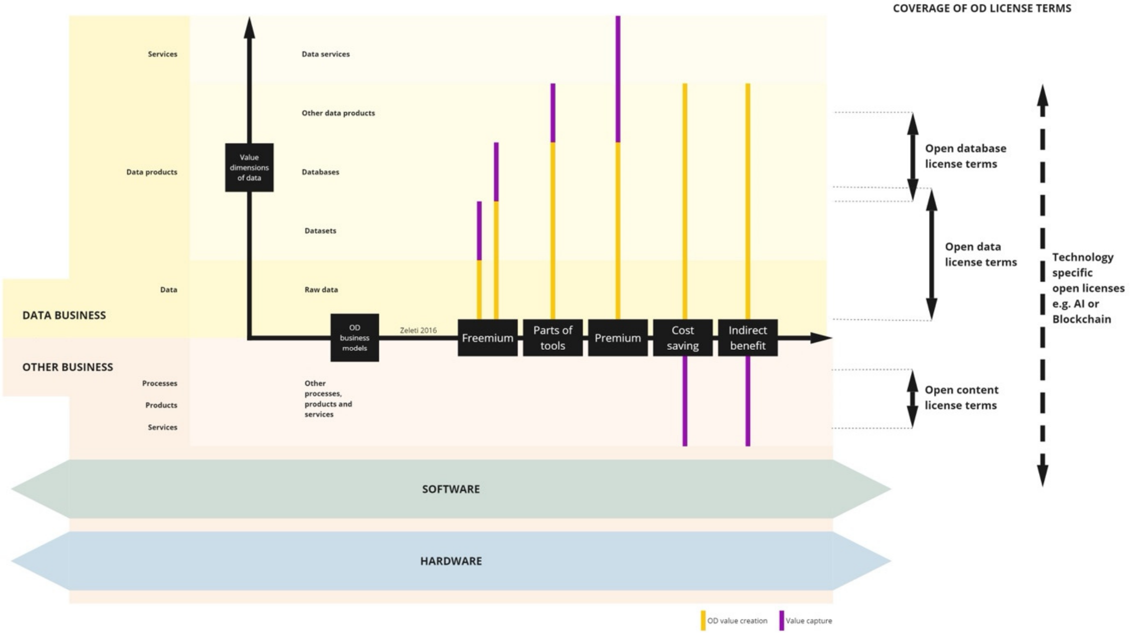


Figure 13. Analysis of the value dimensions of OD business model categories and coverage of concurrent property-focused OD licence terms

We have addressed similar questions from the perspective of FOSS and OD earlier to observe similarities and differences between 3Os and to be able to start addressing the needs from the perspective of hybrid open assets.

From the visualisation, we can first observe the results of the comparison of the open data business model categories to the value dimensions. We can see how different business model categories generate business on different levels of the value dimensions of data. The value dimensions of data consists of levels starting from raw data, continuing to datasets and databases, and ending with data products and data services. The open data business model categories contain open data offerings (value creation aspects) and business offerings (value capture aspects) in various combinations. Secondly, we can detect a clear separation between the value dimensions of data and the current licence terms coverage. The current property-focused licences do not seem to cover consistently the core of the business models and the interfaces between different levels of the value dimensions. Therefore, our proposal is to focus on developing value-focused licensing models for data and, additionally, pay special attention to hybrids of open data based business and other types of business, including but not limited to business based on FOSS and OH or services relating thereto.

As a final comment relating to the challenges ahead with open data licensing, one of the core aspects in data licensing revolves around restrictions set out in the OD and OSS licences. To adhere to the OKFN Open Definition, data needs to be open for anyone and only subject to the acceptable conditions stated in the definition. Such definition is in line with the OSI Open Source Definition[124] regarding open source software and OSHWA Definition[125] regarding open hardware. The restrictions set out in data licences in more restrictive data sharing contexts than purely open data are contradicting the OKFN Open Definition, as these licences may set restrictions as to the role of the data user or as to the forms and context of the use.

These restrictions, however, may also give answers to resolving the difficult question of how to build openness to emerging hybrid technologies – such as in AI – that embrace both source code and data. This aspect is addressed in more detail in Section 4 below.

[124] Open Source Initiative (n 3).

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

3.3.4. Social aspects

Open data, characterised by the transparent sharing of data and the use of technology to disseminate data, has been identified as having social impact[126]. This Subsection examines the social impact of open data by considering both its direct and indirect effects. In addition, this Subsection discusses the current state of measurement of the impact of open data.

The social impact of open data culminates in changes in government and citizens. In fact, it has been recognized that open government data can provide immense value to our societies by educating and providing information to citizens so that they are able to make well-informed decisions, by promoting direct civic engagement and increasing citizens' participation in democratic processes, by gathering feedback for policy makers and the private sector, and by monitoring and holding public officials and the private sector accountable[127]. These are some of the reasons also behind the European strategy for data[128] that aims to make the EU a leader in data-driven society.

Although open data has been identified as having positive social impact, it remains unclear to what extent open data and government transparency have a social impact. In general, it has been recognized that the impact of open data cannot be determined by simple cause-and-effect relationships, as most often the change they achieve is driven by the contribution they make to a complex ecosystem of stakeholders[129]. The ecosystemic aspects and multifaceted value-creation within such ecosystems have been addressed in more detail in ZOOOM D2.3.[130] Due to aforementioned hurdle, the social impact generated by open data is found to be challenging to measure, which is further reflected in methodological challenges around comparability and unevenness of evidence[131]. It has been recognized that the lack of convincing evidence is partly due to the relative newness of open data approaches, but also to difficulties in measuring good governance and social change.

[126] Júlia Keserű and James Kin-Sing Chan, 'The Social Impact of Open Data.' (Sunlight Foundation 2015) <https://www.researchgate.net/publication/298646716_The_Social_Impact_of_Open_Data#fullTextFileContent>.
[127] ibid.
[128] COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS A European strategy for data 2020
[129] Keserű and Chan (n 126).
[130] Ivo Emanuilov and others (n 7).
[131] Keserű and Chan (n 126).

Despite the challenges, the impact of open data is being measured. Yet the current state of the measurement of the impact of open data is still based on quantitative data that is collected from sources such as user statistics, download volumes, number of reuse cases on open data portals and websites[132]. Across the majority of academic articles examining the impact of data, reference is made to existing open data frameworks, the most commonly represented being the Open Data Maturity (ODM) and the Open Data Barometer (ODB)[133]. These two frameworks distinguish the various impacts, and both refer to social impacts. In the current context, these are the first means of measuring the social impact of open data.

While this form of quantitative data has taken steps forward in measuring the impact of open data by providing indications of what the impacts are, it is still unclear how the impacts are actually generated[134]. It is defined that the combination of user statistics and identification of the impact of use cases could provide a better basis for identifying the impact of open data. However, in order to do this, the literature suggests that data of the use cases, including financial data, should be evaluated, which is still a challenge today as many commercial open data users keep their data as their own information because it may create a competitive advantage for competitors[135]. To overcome such hurdles, the companies should have more clarity on the potential impact that opening data may entail and the role the other participants of the ecosystem have in generating such impact.

Thus, it can be concluded that the existing literature suggests that the challenge in assessing the social impact of open data is linked to the lack of sufficient data. Nevertheless, new perspectives for assessing the social impact of open data have been identified. According to Keserű and Chan (2015)[136], a method to measure the social impact of open data requires a shift in perspective towards a focus on medium-term outcomes over long- and short-term outcomes. Even though Keserű and Chan (2015)[137] emphasises the link of normative position and the challenge of defining it to social impact assessment, they believe that by changing the time horizon and using the Outcome Mapping approach to measure the social impact of open data, the right steps towards measuring the social impact of open data can be taken.

[132] Publications Office of the European Union., *Rethinking the Impact of Open Data: A First Step towards a European Impact Assessment for Open Data.* (Publications Office 2023) <https://data.europa.eu/doi/10.2830/911822> accessed 23 September 2023.
[133] ibid.
[134] ibid.
[135] ibid.
[136] Keserű and Chan (n 126).
[137] ibid.

4. ZOOOMing into open source AI

In recent years, the ground-breaking convergence of artificial intelligence (AI) and Free Software (also known as Open Source Software) has fostered exponential accessibility, adoption, reuse and further collaboration of AI-powered technologies, making them more accessible, customizable, and sustainable. The collaborative nature of Free Software projects has nurtured the growth of AI communities, encouraging knowledge sharing, peer review, and collective problem-solving on a global scale. As AI continues to become an integral part of social daily life, ensuring safety, transparency, fairness, and accountability in its development becomes paramount.

In this chapter, we first explore the various challenges surrounding transparency, fairness, and accountability posed by the use of AI systems globally and how Free Software aids in solving the problem. We discuss the proposed regulatory framework for AI in the European Union (EU) and how that might impact the Free Software community. We will then uncover some of the challenges posed by the interplay of AI and Free Software beginning with the latest Free Software adoption trends in the AI domain. This shall be followed by the ethical dimensions of the convergence of AI and Free Software, examining the challenges and opportunities in building responsible and ethical AI solutions within the open source ecosystem. Furthermore, we propound the problem of the use of publicly available Free Software licensed codes for training AI systems particularly in light of the GitHub Copilot case. In line with the discussions on the challenges posed by AI, we briefly delve into the major lawsuits filed on the ground of copyright violations caused by AI systems. Lastly, we will discuss the strides made in addressing these concerns and how the Free Software ethos fosters an environment conducive to ethical, responsible and law-abiding AI advancements.

4.1. Free Software licences and transparency in AI

The proliferation of AI technologies has the potential to transform our societies for good. AI released as Free Software helps in fostering innovation globally as it makes both research and development reproducible and interoperable, without requiring to reinvent the wheel. However, as with every disruptive technology, AI systems come not only with benefits but also with substantial risks, raising a broad variety of legal and ethical challenges. The uncertainty and opacity projected by AI systems is becoming increasingly problematic for the society at large as it may lead to grave discrimination and human rights violation.

Some recent examples of the AI “black-boxes”[138] deployed in the public sector causing imminent risk to human lives include the Dutch welfare fraud detection system, SyRI.[139] In 2015, the Dutch government used the legal instrument, System for Risk Indication (Systeem Risico Indicatie / “SyRI”) to prevent and combat fraud in the field of social security and income-related schemes, tax and social insurance contributions and labour laws. SyRI was used to process large amounts of data collected by various Dutch public authorities to identify those most likely to commit benefits fraud, without any transparency mechanisms or any information about the data used by the algorthim. This led to privacy violations.

In the same vein, in the US, the well-known COMPAS[140] risk assessment tool used a proprietary algorithm to determine the risk of recidivism of criminals. It was revealed that COMPAS judged black and white prisoners differently. It was found that black defendants were far more likely than white defendants to be incorrectly judged to be at a higher risk of recidivism.[141] The Court in this case acknowledged that the “proprietary nature of COMPAS” prevents the disclosure of how risk scores are calculated and criticised the tool for its lack of transparency and the risk for discrimination.[142] Another case relates to the automated fraud detection computer application implemented by the State of Michigan's Unemployment Insurance Agency's (UIA) to determine whether the claimants committed fraud, and execute collection proceedings. The inherent flaws built into the system labelled several people as fraudsters, and collected fines and penalties, all without notice and an opportunity to be heard. Although the case was dismissed for want of subject matter jurisdiction, the AI black box phenomenon affecting human lives could not go amiss.[143]

[138] The notion of black-box AI refers to scenarios in which we can see only input data and output data for algorithm-based systems without understanding exactly what happens in between. For more information, see Warren J von Eschenbach, ‘Transparency and the Black Box Problem: Why We Do Not Trust Ai’ (2021) 34 Philosophy and Technology 1607.

[139] NJCM, Platform Bescherming Burgerrechten & Ors. v. The Netherlands (The SyRI case), [2020] C-09-550982-HA ZA 18-388, Rechtbank Den Haag (District Court of Hague).

[140] COMPAS is an abbreviation for “Correctional Offender Management Profiling for Alternative Sanctions”.

[141] See Jeff Larson Mattu Julia Angwin,Lauren Kirchner,Surya, ‘How We Analyzed the COMPAS Recidivism Algorithm’ [2023] *ProPublica* <https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm> accessed 27 September 2023.

[142] State v Loomis, [2016] 881 NW2d 749 (2016), Supreme court of Wisconsin.

[143] Cahoo et al. v. SAS Analytics Inc.,[2020] 17-10657, 08-11-2020, United States District Court, E.D. Michigan, Southern Division.

Similarly, in Australia, a ‘Robodebt’ algorithm automated debt assessment and recovery which was deployed by Services Australia as part of its Centrelink payment compliance program resulted in 20,000 people being falsely accused of fraud. The Australian government in November 2020, agreed to a 1.2billion dollar settlement for a class action brought on behalf of Robodebt victims.[144]

The above examples illustrate that transparency is imperative to control, monitor and correct AI systems. The primary components of transparency are accessibility and comprehensibility of information, which includes the information about functionality of the algorithms.[145] Against this background of use of AI systems in public administration, when an AI software is distributed under a Free Software licence, its potential risks can better be identified and avoided. A Free Software licence can make the AI systems more auditable, thus inspiring more transparency, trustworthiness and safety for everyone. This will in turn enhance the promotion and uptake of these technologies. Furthermore, AI released under a Free Software licence fosters innovation and competition globally owing to the freedoms that Free Software offers (to use, to study, to improve, to share).

In line with this position, the “Public Money? Public Code!” initiative, launched by the Free Software Foundation Europe (FSFE), demands that publicly financed software must be made publicly available under a Free Software licence.[146] Publicly financed AI technologies also fall under this scope. The initiative encourages that public procurement should support Free Software AI and digital solutions to be used in the public, but also private sector. Public authorities using AI systems should publish those, including a description, version history and the source code in a public register. Free Software is rooted in openness and transparency, and its development is based on collaborative networks of programmers and stakeholders. Therefore, the need for public software platforms (code repositories) in which the code and also best practices are shared is a must. These platforms will enhance sharing solutions as well as improving auditability by making the identification of vulnerabilities easier and, consequently, faster to tackle and fix them. Furthermore, if an AI system uses open data, the corresponding code must also be made publicly findable and accessible under a Free Software licence. Thereby all stakeholders, not only developers, can

[144] Luke Henriques-Gomes, ‘Robodebt Class Action: Coalition Agrees to Pay $1.2bn to Settle Lawsuit’ *The Guardian* (16 November 2020) <https://www.theguardian.com/australia-news/2020/nov/16/robodebt-class-action-coalition-agrees-to-pay-12bn-to-settle-lawsuit> accessed 27 September 2023.

[145] Brent Daniel Mittelstadt and others, ‘The Ethics of Algorithms: Mapping the Debate’ (2016) 3 Big Data & Society 2053951716679679.

[146] See more at: https://publiccode.eu/

contribute and discuss future developments in furtherance of achieving the highest potential of AI technologies.[147]

4.2. AI regulatory initiatives

From a regulatory perspective, on April 21, 2021, the European Commission presented its proposal for a Regulation "laying down harmonised rules on Artificial Intelligence", the so-called "Artificial Intelligence Act" (AIA).[148] In the context of this proposed legislation, there are several deficiencies vis-à-vis Free Software that need to be addressed in order to establish a future-proofed legislation on AI. The EU is currently aiming to introduce liability rules for software, which can affect Free Software. Although the main debates revolve around the Cyber Resilience Act (CRA) and the Product Liability Directive (PLD), similar rules are being introduced in the AI Act.[149]

Although the AIA is still not in force, the EU Parliament has adopted the final text, including a far-reaching exemption for non-profit organisations and small Free Software projects up to the size of micro-enterprises. Negotiations on the final text between the three institutions (also known as trialogue, where representatives of the Parliament, the Council, and the Commission meet to agree on a final text) are already taking place. Here the FSFE calls on the EU Parliament to defend its position to ensure that this regulation protects not only Free Software and its contributors but also consumers and customers. The main debate happens around the Cyber Resilience Act, so the FSFE has proposed a strategy using this particular legislation as a departing point. While the FSFE welcomes the discussion on more cybersecurity the introduction of liability rules alone won't necessarily lead to more safety.[150] Especially in Free Software, far-reaching security measures can be already put in place, differently from those of proprietary software. The proposal to exclude Free Software "outside the course of a commercial activity" would fail to address a large part of software that will not be covered but is deployed. At the same time smaller and non-profit projects would be harmed as they would have to bear major costs. Therefore,

[147] FSFE, "Artificial Intelligence and Free Software (also known as Open Source Software)". 2022. Available at: https://download.fsfe.org/campaigns/AIandFS/fsfe_AIandFreesoftware.pdf

[148] On December 6, 2022, the Council of the EU adopted its general approach and compromise text on the proposed Regulation Laying Down Harmonized Rules on Artificial Intelligence (the "AI Act"). More at: https://eur-lex.europa.eu/legal-content/EN/HIS/?uri=CELEX:52021PC0206

[149] FSFE, "Liability, Interoperability & Free Software in EU: what we are expecting". 24.08.2023. Available at: https://fsfe.org/news/2023/news-20230824-01.html

[150] FSFE, "EU: Proposed liability rules will harm Free Software". Available at: https://fsfe.org/news/2023/news-20230323-01.html.

from the standpoint of responsibility and liability, in order to enhance security while safeguarding the AI ecosystem, responsibility and liability should be shifted to:[151]

1. Those deploying Free Software instead of those developing Free Software and;
2. Those who significantly financially benefit from this deployment should make sure the software is compliant with the rules provided under the AIA.

This is critically important for AI innovation and levying any impositions on it would have a chilling effect on open collaboration in the AI ecosystem. Secondly, an existing algorithm that can be reused benefits the whole research ecosystem. However, the AIA does not provide for a research exemption unlike few other regulations[152]. If a researcher or developer collaborates with the industry and publishes their model for academic purposes, they may run the risk of being regarded as providers who "develop an AI system" with a view of "putting it into service" (Art. 3(2) AIA), ie, supplying the system "for first use directly to the user or for own use" (Art. 3(11) AIA). This provision should be modified to clarify that the obligations of a provider should not be triggered in the course of general AI research or when publishing a paper.[153] In this regard, the EU Parliament has made positive advancements by adopting its final text protecting Free Software with a large majority. It includes a far-reaching exemption for non-profit organisations and small Free Software projects up to the size of micro-enterprises. This will be followed up by a trilogue with the EU Parliament, Council and Commission on the final text.[154]

While the AIA is not yet in force, there are also a myriad of other open-ended questions and concerns that need to be addressed along the same lines. The section below highlights various problems at the intersection of AI and Free Software.

[151] FSFE, "EU: Proposed liability rules will harm Free Software". Available at: https://fsfe.org/news/2023/news-20230323-01.html

[152] An example of that would be Art. 89 of GDPR. See more inNathalie A Smuha and others, 'How the EU Can Achieve Legally Trustworthy AI: A Response to the European Commission's Proposal for an Artificial Intelligence Act' (5 August 2021) <https://papers.ssrn.com/abstract=3899991> accessed 27 September 2023.

[153] ibid.

[154] See, FSFE, "EU Parliament wants to protect Free Software in AI regulation", 11.05.2023, available at: https://fsfe.org/news/2023/news-20230511-01.html ; Free Software Foundation Europe, "EU: Majority for AI Act – and safeguards for Free Software", 14.06.2023. https://fsfe.org/news/2023/news-20230614-01.html. See also the Amendments to the Artificial Intelligence Act adopted by the EU Parliament on 14.06.2023, available at: https://www.europarl.europa.eu/doceo/document/TA-9-2023-0236_EN.html

4.3. Challenges at the intersection of AI and free software

4.3.1. Free Software adoption trends in the AI domain

In this section we analyse the current trends on licensing AI-based technology from the perspective of Free Software. We provide an overview on the challenges such models impose on Free Software and how this can impact compliance. Particular attention is given to licensing models imposing additional behaviour restrictions, and how such limitations would interact with Free Software principles.

4.3.2. The Rights Ratchet Model

Although not a new phenomenon, the "rights ratchet model" has also spilled over the AI domain. This model refers to the practice whereby a company initially uses a Free Software licensing model and when the adoption of its product grows exponentially, owing to the freedoms of Free Software licences, it slowly starts reducing the openness and freeness. Typically, these companies use a Contributor Agreement despite using a Free Software licence in order to give the company the rights to their code and harvest community copyrights.[155] Subsequently, these companies try to change the licence in order to focus on how to best monetize work without considering the commitments to the Free Software communities from whom they benefited significantly.[156] This model takes about 10 years of transition time from completely Free Software to proprietary and uses Free Software licences only as an initial development model as opposed to its entire life cycle.[157]

A case in point is the Open AI. The company's founding ethos were based on a Free Software model of development when it was founded in 2015. It started as a non-profit but later became a "capped profit" in order to secure billions in investment, primarily from Microsoft, with whom it now has exclusive business licences.[158] GPT-2, the first

[155] See, 'Rights Ratchet' (*Meshed Insights Ltd*, 15 September 2022) <https://meshedinsights.com/tag/rights-ratchet/> accessed 27 September 2023.

[156] Ibid; See also, Steven Vaughan-Nichols, "Open Source Initiative expands its role to AI and machine learning", ZDNET, Sep 2022, available at: https://www.zdnet.com/article/open-source-initiative-expands-its-role-to-ai-and-machine-learning/.

[157] Ibid.

[158] James Vincent, 'OpenAI Co-Founder on Company's Past Approach to Openly Sharing Research: "We Were Wrong"' (*The Verge*, 15 March 2023)

model of their Large Language Model (LLM) released by OpenAI, was not released and shared due to concerns about the ability of these models to "generate deceptive, biased, or abusive language at scale".[159] The next version of the model, GPT-3, was made available only through a permissioned API and in parallel preferential use of the technology as exclusively licensed to Microsoft.[160] Subsequently, in their technical report releasing GPT-4 in March 2023, Open AI have cited "security" and "competition" as the reasons to maintain a "closed off" approach.[161]

It's interesting to observe that while there may be a host of reasons cited to adopt a "closed off" approach by the companies who originally stood by the principles of Free Software; the rights ratchet model is increasingly becoming the modus operandi especially to harness the immense potential of AI technologies.[162] This poses serious considerations on the principles of Free Software.

4.3.3. Usage of licences with additional restrictions

In the context of steady emergence of AI systems, the bedrock of societies, communities, economies and sustainable development hinges on how technology is transposed into societies. The several underpinnings of these technologies include technological feasibility, legal compliance, and socio-ethical acceptability. On the one hand, technology feasibility as an economic indicator provides a unidimensional quantitative value of market-driven innovation of AI;[163] socio-ethical acceptability is multi-dimensional that enables "responsible AI" on the other hand. While social issues give rise to policy solutions and thus legislations; ethics are deeply rooted in societal values that differ greatly from one jurisdiction to another. Therefore, they need to be carefully embedded in technologies, given the consequences of their application to AI

<https://www.theverge.com/2023/3/15/23640180/openai-gpt-4-launch-closed-research-ilya-sutskever-interview> accessed 27 September 2023.

[159] See, https://openai.com/research/better-language-models , Feb 2019.

[160] See, https://openai.com/blog/openai-licenses-gpt-3-technology-to-microsoft, Sep 2020.; see also Alek Tarkowski, 'Notes on BLOOM, RAIL and Openness of AI' [2022] Open Future <https://openfuture.pubpub.org/pub/notes-on-open-ai/release/1> accessed 27 September 2023.

[161] See, OpenAI (2023), 'GPT-4 Technical Report' (OpenAI) s 2 <https://cdn.openai.com/papers/gpt-4.pdf>.

[162] For instance, Open AI, as discussed above; Similarly, Lightbend recently dropped Akka's (an important open-source Java middleware program) Free Software license from Apache 2.0 to non-free Business Source License (BSL) 1.1. More information available at: https://www.lightbend.com/blog/why-we-are-changing-the-license-for-akka

[163] Montreal AI Ethics Institute, The State of AI Ethics, June 2020.

systems. Problems of transparency, accessibility, explainability and interpretability make the deployment of AI systems ethically challenging.[164]

In this regard, there are several AI ethics guidelines which have laid down a framework for fostering and securing ethical and robust AI such as The Ethics Guidelines For Trustworthy AI propounded by the High-Level Expert Group on Artificial Intelligence[165], the Montreal Declaration for Responsible AI.[166] However, the adaptation of such guidelines remains voluntary. Therefore, there is a need for auditing of AI or laying down standards. With reference to this, scholars have highlighted a range of different approaches to ethics-based auditing of AI that already exists. Functionality audits focus on the rationale behind the decision, code audits entail reviewing the source code, and impact audits investigate the effects of an algorithm's outputs.[167] Interestingly, using Free Software licences enables such code audits by allowing any user to not only review the source code but also improve it, thus aiding better explainability of AI systems.

In the last decade, diverse groups and individuals have departed from using exclusively Free Software licences on their projects to develop new types of licensing solutions which prioritise determined restrictions on how software should be further used and distributed. Such licensing schemes emphasise restrictions related to fields of endeavour, behaviour, community management and commercial practices. As an example, the failed attempt of the Exception General Public License (eGPL) in 2009,[168] forbade military users and suppliers from using its code. In 2019, Coraline Ada Ehmke, founder of The Organization for Ethical Source (OES)[169] created the Hippocratic License which is based on the MIT licence with a modified clause. This licence bans uses for certain purposes:

"The software may not be used by individuals, corporations, governments, or other groups for systems or activities that actively and knowingly endanger, harm, or otherwise threaten the physical, mental, economic, or general well-being of individuals or groups in violation of the United Nations Universal Declaration of Human Rights."

[164] Mittelstadt and others (n 145).
[165] 'High-Level Expert Group on Artificial Intelligence | Shaping Europe's Digital Future' (27 September 2023) <https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai> accessed 27 September 2023.
[166] Montreal Declaration for Responsible AI, available at: https://recherche.umontreal.ca/english/strategic-initiatives/montreal-declaration-for-a-responsible-ai/.
[167] Jakob Mökander and Luciano Floridi, 'Ethics-Based Auditing to Develop Trustworthy AI' (2021) 31 Minds and Machines 323.
[168] http://blog.egpl.info/about/
[169] See, Organization for Ethical Source, available at: https://ethicalsource.dev/what-we-do/.

At the outset, the terms of the licence contravenes the four freedoms (use, share, study and improve) provided according to the definition of Free Software as well as the Open Source definition as Open Source licences may not “restrict anyone from making use of the program in a specific field of endeavour.” The OSI did not approve the Hippocratic licence as an Open Source License.[170]

In 2021, the OES released the Hippocratic License 3.0 (HL3),[171] a major revision of the preeminent ethical source licence that specifically prohibits the use of open source software in violation of universal standards for human rights. The core licence provides protections for universally recognized human rights as enshrined under United Nations Universal Declaration of Human Rights, the International Covenant on Economic, Social and Cultural Rights, the International Covenant on Civil and Political Rights, and the International Labour Organization. HL3 also offers optional modules that focus on specific areas of concern, such as environmental justice, labour rights, and ethical supply chains of their particular communities. Despite the reference to the various legal treaties, the actual efficacy of the licence is highly questionable due to the fact that the interpretation of the framework of human rights is still contentious from the lens of emerging technologies. What can be termed as a “harm” is far from straightforward especially in light of algorithmic activities where the harm is hard to debug owing to the black-box phenomenon of AI and thus, so is the attribution of the harm itself. Moreover, a mere licence cannot substitute the law enforcing agencies in ensuring the “ethical” use of software. Therefore, these restrictive licences should not cause to overstep the process of legitimate legislative process and substitute any regulation or legal order.

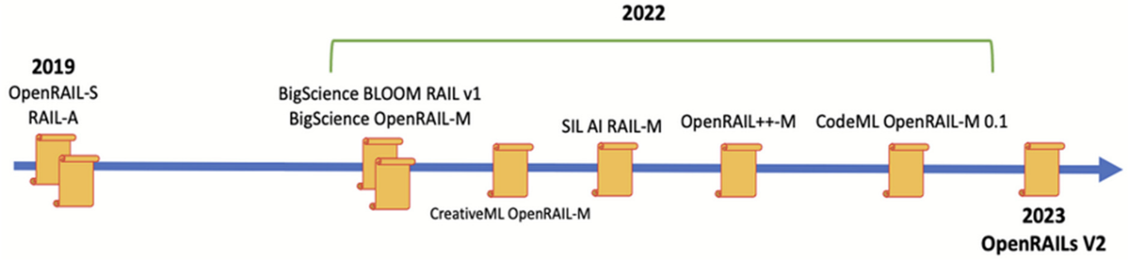
Another licence that concerned the ethical labour practices was the Anti 996 licence inspired by China's burgeoning anti-996 movement. It required users to comply with both local and international labour standards as provided by the International Labour Organization, in response to reports of gruelling working conditions at Chinese technology companies. While there are no definite figures supporting the actual uptake of the licence by various AI projects, it certainly did gain international

[170] On September 23rd, the OSI responded (via Twitter) complained to the release of the Hippocratic License, that the “intro to the Hippocratic Licence might lead some to believe the license is an Open Source Software licence, and software distributed under the Hippocratic Licence is Open Source Software. As neither is true, we ask you to please modify the language to remove confusion.”
[171] See, Hippocratic License 3.0 (HL3): An Ethical License for Open Source Communities, available at: https://firstdonoharm.dev/

traction.[172] Despite the international support towards the cause, it failed to align with either the "Free Software" or "Open Source" definitions.

Within the private sector, companies such as IBM, Microsoft and Google's DeepMind have joined forces in creating broad initiatives such as the 'Partnership on AI'[173] or 'OpenAI'[174], which bring together industry, non-profit and academic organisations. The same companies have also established their own ethic codes on AI in the form of Responsible AI License (RAIL) that offers end user licence agreements and source code licence agreements which creates a broad range of behavioural (or use-based) restrictions.

The RAIL initiative[175] as created in 2019 to encourage the industry to adopt use restrictions in licences as a way to mitigate the risks of misuse and potential harm caused by AI systems. RAILs can be used to licence data (D), Apps (A), models (M), and source code (S). depending on the AI feature(s) you are licensing, you will add suffix D, A, M, or S.[176]



The table above gives an overview of the evolution of the various RAILs.[177]

- Responsible AI End-User License - this regulates the way in which ready-to-function AI is used as a whole, without the possibility of it being modified in any.

[172] Microsoft Workers, 'Microsoft and GitHub Workers Support 996.ICU' <https://github.com/MSWorkers/support.996.ICU> accessed 28 September 2023.
[173] Available at: https://www.partnershiponai.org/
[174] Available at: https://openai.com/
[175] Available at: https://www.licenses.ai/
[176] More information can be obtained here: https://www.licenses.ai/blog/2022/8/18/naming-convention-of-responsible-ai-licenses
[177] The information in the table was kindly derived from The Responsible AI License FAQ section, available at: https://www.licenses.ai/faq-2. Responsible AI, 'AI Licenses' (*Responsible AI Licenses (RAIL)*) <https://www.licenses.ai/ai-licenses> accessed 24 September 2023.

- Responsible AI Source Code License – this instrument regulates the way in which an AI source code is used, including any possible scope for its modification.[178]

The clauses in both these agreements, creating restrictions and obligations owed to the licensor (such as prohibiting any infringement of third-party intellectual property rights), are designed from an “ethical” standpoint. The RAIL licence has an appendix list that contains the various prohibited uses, allowing different communities to choose different prohibitions.

RAIL claims to be “open” as it enables reuse, distribution, commercialization, and adaptation as long as the artefact is not being applied for use-cases that have been restricted.[179] However, this is in diametric opposition to the four freedoms provided by Free Software, namely, freedom to use, study, share and improve. In addition, this is also in contravention to the definition of “open source” as defined by the Open-Source Initiative.

Secondly, the Open RAIL-M provides different communities to choose specific prohibitions for the use of the models.[180] However, this leads to serious interoperability issues if two models containing different obligations are combined together.

Use case study of RAIL: BLOOM is an acronym for BigScience Large Open-science Open-access Multilingual Language Model. As the name suggests, it’s a large language model that was created over the last year by over 1,000 volunteer researchers in a project called BigScience, which was coordinated by AI startup Hugging Face and is co-funded by the French government. Touted to be Open Science and Open source, it uses the RAIL licence. One of the conditions provided in the licence is that it prohibits the use of BLOOM in areas such as law enforcement, healthcare, or deception. This, as posited before, fundamentally goes against the four freedoms provided by Free Software, namely, freedom to use, study, share and improve. Moreover, the criteria of “No Discrimination Against Fields of Endeavor” as well as “Free Distribution” required to label a software as “open” as per the definition laid down by the Open-Source Initiative is also violated by the use of the RAIL License.

[178] See Kamil Szpyt, ‘Responsible AI Licenses- a Real Alter- Native to Generally Applicable Laws?’ (2020) 1 Revista Ibérica Do Direito 178.
[179] See, sec III of BigScience RAIL License v1.0 at https://huggingface.co/spaces/bigscience/license
[180] See Appendix of the Open RAIL-M license available at: https://www.licenses.ai/blog/2022/8/26/bigscience-open-rail-m-license.

In the same vein, Meta's OPT-175B model is released under a licence[181] that limits uses to non-commercial, research uses. It includes additional use restrictions covering biometric processing, nuclear technologies, and any military or surveillance purposes.

Broadly speaking, RAIL can also be perceived as potentially fragmenting the Free Software community by introducing a new licensing framework that may not be compatible with existing licences. This can create challenges when trying to combine RAIL-licensed code with other Free Software projects. A primary example can be drawn from the fact that in a jurisdiction agnostic software development community, Free Software enables collaboration among different stakeholders owing to the freedoms that it provides that have been clearly defined, leaving no room for any ambiguity. However, the subject of "ethics" is significantly broad and hosts a wide spectrum of interpretations, especially adjudging from an international perspective.

Secondly, the list of use restrictions in RAILs is not conceivably exhaustive and the choice of use restrictions admittedly depends on the licensor, which is usually based on the awareness of the technical capabilities and limitations of an AI artefact. This creates challenges in determining implementation of these licences as well as its compliance. The proliferation of licences with behavioural restrictions, principally lacks the legal precedence and clarity found in more established Free Software licences which can lead to potential ambiguity and conflicts. While RAILs has been more widely recognized in comparison to some of the above mentioned restrictive licences, the community support and adoption is gravely low in contrast to the Free Software licences.[182]

Besides, such licences may pose challenges to comply with obligations under the EU legislation, affecting regulatory safeguards. For example, the CodeML Open RAIL-M v0.1 License's Use Restrictions provides for the following:

> *"You agree not to use the Model or Derivatives of the Model:*
>
> *….*
>
> *(e) To generate or disseminate personal identifiable information that can be used to harm an individual;*

[181] Available at: https://github.com/facebookresearch/metaseq/blob/main/projects/OPT/MODEL_LICENSE.md

[182] See study conducted by Paul Keller and Nicolò Bonato, 'Growth of Responsible AI Licensing. Analysis of License Use for ML Models Published on 🤗' [2023] Open Future <https://openfuture.pubpub.org/pub/growth-of-responsible-ai-licensing/release/2> accessed 28 September 2023.

> *(i) For fully automated decision making that adversely impacts an individual's legal rights or otherwise creates or modifies a binding, enforceable obligation; …"*

While GDPR defines as to what "personal data"[183] could entail, no reference to the specific legislation highlighting the definition or obligation thereunder is alluded to. Furthermore, the concept of "harm" can be expounded by way of different legislation in a different light. Similarly, clause (i) of the CodeML Open RAIL-M v0.1 License seemingly corresponds to Article 22 of GDPR, however, without reference to its limitation and exclusion clauses, the clause in CodeML Open RAIL-M v0.1 does not align itself to the mandate of the law. In the absence of any real correspondence to the regulations, these restrictive licences may only reduce to a tool for self-regulation based on community norms.

Equally important would be to discuss the consequences of violation of the mentioned contents of these restrictive licences (case in point, CodeML Open RAIL-M v0.1) facing the proportionality principle regarding licence infringement. Determining the "ethical compliance" face of "legal compliance" can be challenging when all other licence conditions are being met.[184]

While the initiative to have a Responsible AI aligns with all the policies documented in this domain, licensing frameworks like RAIL does not qualify as "open" or "free" whilst driving the initiative. Moreover, given the various obstacles in using these restrictive licences, ranging from technical, legal to compliance, it would be appropriate to rather deploy tools that further fairness, explainability, and security of the applications and systems. Few of these tools are summarised below are powered by the LF AI & Data Foundation:[185]

- AI Fairness 360:[186] This extensible open-source tool kit helps users to examine, report, and mitigate discrimination and bias in machine learning models throughout the AI. It is licensed under Apache 2.0.
- AI Explainability 360:[187] This open-source library supports the interpretability and explainability of datasets and machine learning models. It is licensed under Apache 2.0.

[183] Article 4(1) of GDPR.

[184] See in particular, Szpyt (n 178). See also Luis Villa, "Evaluating the RAIL license family", Tidelift,

[185] See, https://community.linuxfoundation.org/lf-ai-data-foundation/. See, in general, Ibrahim Haddad, 'Artificial Intelligence and Data in Open Source' (The Linux Foundation 2022) Research <https://www.linuxfoundation.org/research/artificial-intelligence-and-data-in-open-source> accessed 28 September 2023.

[186] For more details, see https://ai-fairness-360.org.

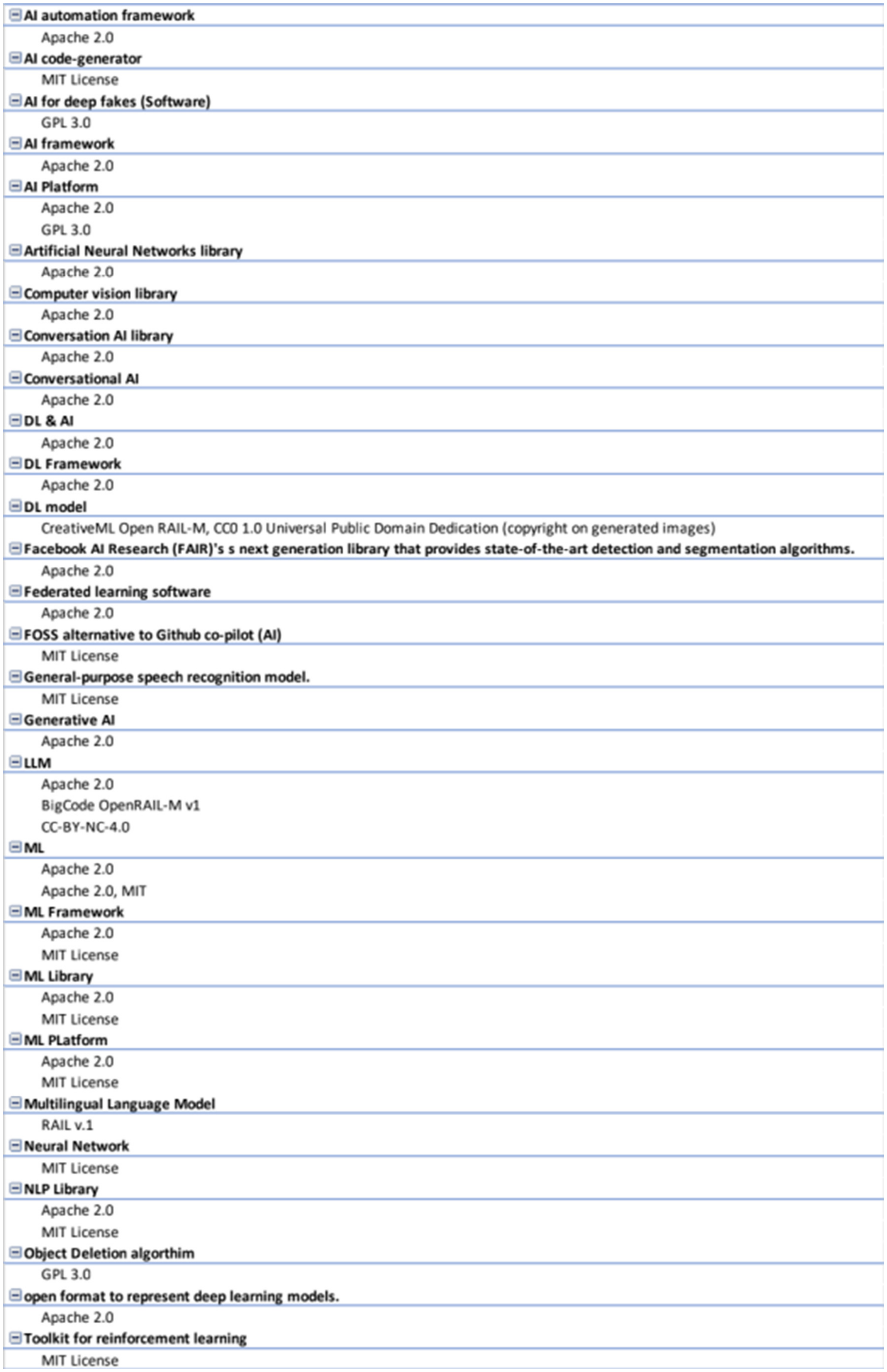
[187] For more information, see https://ai-explainability-360.org/.

- Adversarial Robustness Toolbox (ART):[188] This open-source tool helps developers and researchers to evaluate, defend, and verify machine learning models and applications against adversarial threats. It is licensed under MIT.

4.4. Usage of free software licences in the AI domain

The following section first provides a quantitative analysis on the kinds of free software licences commonly used in the AI domains and subdomains. Against this background, it analyses whether these licences actually align with the definition of “Free Software”. Lastly, the implications of restrictive behaviour of these licences on the society at large is discussed.

[188] For more information, see https://adversarial-robustness-toolbox.org/.



- ⊟ **AI automation framework**
  - Apache 2.0
- ⊟ **AI code-generator**
  - MIT License
- ⊟ **AI for deep fakes (Software)**
  - GPL 3.0
- ⊟ **AI framework**
  - Apache 2.0
- ⊟ **AI Platform**
  - Apache 2.0
  - GPL 3.0
- ⊟ **Artificial Neural Networks library**
  - Apache 2.0
- ⊟ **Computer vision library**
  - Apache 2.0
- ⊟ **Conversation AI library**
  - Apache 2.0
- ⊟ **Conversational AI**
  - Apache 2.0
- ⊟ **DL & AI**
  - Apache 2.0
- ⊟ **DL Framework**
  - Apache 2.0
- ⊟ **DL model**
  - CreativeML Open RAIL-M, CC0 1.0 Universal Public Domain Dedication (copyright on generated images)
- ⊟ **Facebook AI Research (FAIR)'s s next generation library that provides state-of-the-art detection and segmentation algorithms.**
  - Apache 2.0
- ⊟ **Federated learning software**
  - Apache 2.0
- ⊟ **FOSS alternative to Github co-pilot (AI)**
  - MIT License
- ⊟ **General-purpose speech recognition model.**
  - MIT License
- ⊟ **Generative AI**
  - Apache 2.0
- ⊟ **LLM**
  - Apache 2.0
  - BigCode OpenRAIL-M v1
  - CC-BY-NC-4.0
- ⊟ **ML**
  - Apache 2.0
  - Apache 2.0, MIT
- ⊟ **ML Framework**
  - Apache 2.0
  - MIT License
- ⊟ **ML Library**
  - Apache 2.0
  - MIT License
- ⊟ **ML PLatform**
  - Apache 2.0
  - MIT License
- ⊟ **Multilingual Language Model**
  - RAIL v.1
- ⊟ **Neural Network**
  - MIT License
- ⊟ **NLP Library**
  - Apache 2.0
  - MIT License
- ⊟ **Object Deletion algorthim**
  - GPL 3.0
- ⊟ **open format to represent deep learning models.**
  - Apache 2.0
- ⊟ **Toolkit for reinforcement learning**
  - MIT License

Figure 14. Commonly used Free Software licences in each of the AI domains and subdomains[189]

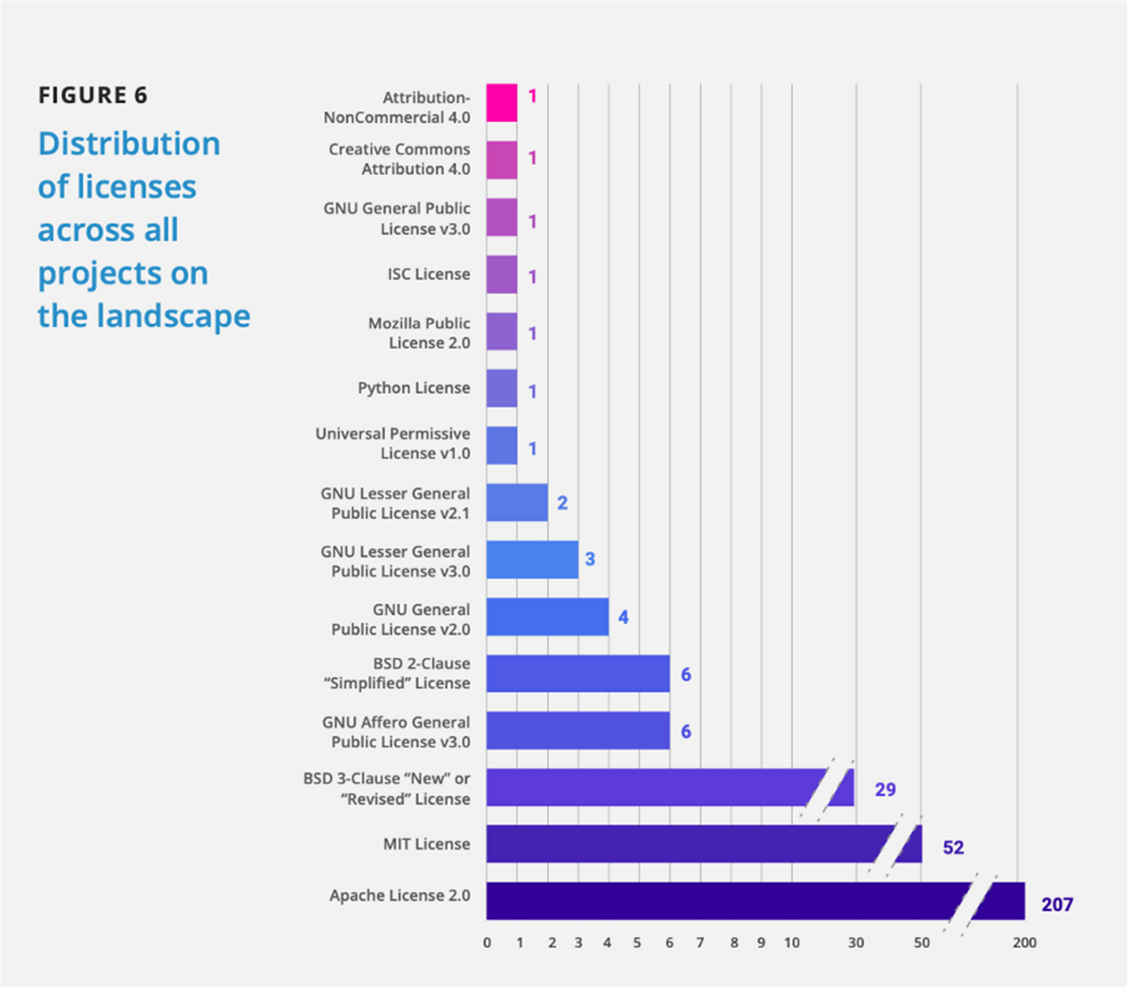


Figure 15. Percentage of uptake of Free Software licences corresponding to AI based projects[190]

The first table represents the most commonly used Free Software licences in each of the AI domains and subdomains. The second table indicates the percentage of uptake

[189] This table is an outcome of a desk research conducted from April, 2023 to August, 2023. Differently online sources were considered, especially large source code repositories such as GitHub and GitLab. Information on licensing was taken from the projects' repositories and main webpages. The research also reviewed and retrieved data from Haddad, I. Dobrin, S. (2022). *Artificial Intelligence and Data in Open Source*. The Linux Foundation. available at: https://www.linuxfoundation.org/research/artificial-intelligence-and-data-in-open-source. This work was also used for categorization of the AI domains and subdomains.

[190] The information in the table was derived and adapted from Haddad, I. Dobrin, S. (2022). *Artificial Intelligence and Data in Open Source*. The Linux Foundation. available at: https://www.linuxfoundation.org/research/artificial-intelligence-and-data-in-open-source. This table corresponds to projects within the LF AI & Data Framework.

of Free Software licences corresponding to AI based projects. It suggests that when it comes to the licensing of AI projects, Apache 2.0 licence is the most preferred Free Software licence, followed by MIT. However, despite the use of these Free Software licences, a common trend that is currently observed in the AI industry is that companies are heading to jump on the Free Software bandwagon but are still imposing certain conditions that have a restrictive character. This renders them as non-Free Software as per the definition of Free Software.

The adoption of AI licences is rapid. As can be seen from *this quote*, Open RAIL has become very rapidly the second most used category right after permissive open source software licences.

> *"According to a recent paper that analyzed licenses attached to models on the HuggingFace hub, between September 2022 and January 2023, Open RAIL licenses have overtaken all other categories of restrictive open source licenses, and are now the second most used category after permissive open source software licenses."*

Growth and Adoption of RAIL Licenses — Responsible AI Licenses (RAIL)

Free Software refers to software that enables its users to maintain their control and freedom over how to use such software. More specifically, the following four essential freedoms define Free Software:

- Freedom to use
- Freedom to study
- Freedom to share
- Freedom to improve

Licences with restrictive characteristics do not align with the above freedoms. In light of the AI developments, any restrictions on the distribution and modification of the software cannot facilitate its use, rather only cause obstruction to its advancement. The lack of advancement can be enunciated by way of three predominant material harms, namely, the following[191]:

[191] Richard Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman* (Free Software Foundation) 124 <https://www.gnu.org/philosophy/fsfs/rms-essays.pdf>.

- **Fewer people use the AI program**: The conditions of the RAIL licences raise many ambiguities in its implementation. For example, the condition prohibiting any intentional defamation, disparagement or harassment to others. Furthermore, the condition to not discriminate against individuals or groups based on legally protected characteristics or categories. The vague terminologies used in these clauses are open to interpretation and thus result in an overall board prohibition to its use in the first place.

- **None of the users can adapt or fix the program**: Unlike the direct licensing model provided by Free Software licences, some of the restrictive licences (for example RAIL) follow the sub-licensing model. In this case, the initial licensor can only enforce licence conditions against the initial licensee and not the subsequent downstream licensees. However, the initial licensee is free to re-license the model under their own licence provided that they flow down the use restrictions in the RAIL licence. Therefore, the downstream users receive fewer rights than upstream users and the software no longer is "free" or "open" for equivalent adaptation.[192] Another example is Stable Diffusion, whose text-to-image foundation model can potentially create accurate medical images leading to better clinical healthcare. However, the use of the CreativeML Open RAIL-M licence[193] has rather created barriers for adoption by preventing users from generating images for medical advice or medical results interpretation.[194]

- **Other users cannot base new work on it:** Meta's LLaMA model can be cited as a case in point. Meta's LLaMA model, a language learning model, forms the basis of many open-source chatbots, including Alpaca, Vicuna, and OpenAssistant.[195] It was originally licensed under GPL v.3.0 model and is available for research purposes but cannot be used commercially. Although it provides the freedom to study and improve, it prohibits the freedom to use and share for any other purpose. As the definition of Free Software and even Open Source does not prohibit any commercial use of a software, the same cannot

[192] Kate Downing, 'AI Licensing Can't Balance "Open" with "Responsible"' (*Law Offices of Kate Downing*, 14 July 2023) <https://katedowninglaw.com/2023/07/13/ai-licensing-cant-balance-open-with-responsible/> accessed 28 September 2023.

[193] License available at: https://huggingface.co/spaces/CompVis/stable-diffusion-license.

[194] See also, 'Stable Diffusion Could Solve a Gap in Medical Imaging Data' (*VentureBeat*, 13 February 2023) <https://venturebeat.com/ai/stable-diffusion-could-solve-a-gap-in-medical-imaging-data/> accessed 28 September 2023.

[195] See Maximilian Schreiner, 'Metas LLaMA v2 Could Shake up the Chatbot Business' (*THE DECODER*, 16 June 2023) <https://the-decoder.com/metas-llama-v2-could-shake-up-the-chatbot-business/> accessed 28 September 2023.

be deemed as a Free Software model.[196] However, there are some indications of Meta allowing commercial use of its LLaMa in a subsequent v.02.[197]

Against this background, given the proliferation of these licences with behavioural restrictions, integration of Free Software, licence compatibility and licensing compliance become more complex, affecting the whole ecosystem. There is a strong need for the companies adopting these licences to re-visit their Free Software licensing framework and align with the principles of the Free Software licences. The issue becomes even more critical where the public money is being used to fund the projects which at some stage exhibit restrictive practices. For instance, the BigScience Project which has launched BLOOM that uses the RAIL licence, is funded by the French National Institute for Scientific Research (CNRS),[198] GENCI and the French Ministry of Higher Education and Research.

4.5. Free software licensing compliance and AI litigation

Free Software has been used to power AI-based technologies. Corporations have implemented source code protected under a Free Software licence in diverse environments and for a large spectrum of solutions. Compliance with licence obligations are paramount to avoid infringement. In this section we analyse recent cases involving Free Software licensing compliance and AI and provide an overview of recent contentious cases regarding copyright violations caused by AI systems. We will not get into the details of the actual application of copyright to AI systems, but refer only to questions relevant to Free Software licensing.

[196] See Open Source Initiative (n 3). Also Free Software Foundation, 'What Is Free Software?' (*GNU Project*) <https://www.gnu.org/philosophy/free-sw.html> accessed 19 February 2023.

[197] Open Source Initiative (n 3).

[198] The CNRS was involved in particular through its Institute for Development and Resources in Intensive Scientific Computing (IDRIS). GENCI (Grand équipement national de calcul intensif) is responsible for promoting high-performance computing in France; See 'Release of Largest Trained Open-Science Multilingual Language Model Ever | CNRS' (*Press Area*, 12 July 2022) <https://www.cnrs.fr/en/press/release-largest-trained-open-science-multilingual-language-model-ever> accessed 28 September 2023.

4.5.1. GitHub Copilot: the use of publicly available Free Software licensed codes for training AI systems

The GitHub Copilot is an AI-based coding assistant trained on publicly accessible open-source licensed code which recently got embroiled in a class action lawsuit in the US District Court for the Northern District of California[199]. This is the first Free Software copyright lawsuit filed challenging the legality of GitHub Copilot and the related ML of OpenAI Codex. The lawsuit against GitHub, Microsoft, and OpenAI claims violation of Free Software licences, many of which are covered by strong copyleft licences (ie, GPL v2, GPL v3) based on which Copilot is trained. Copyleft licences require that derivative works (of the copyleft-licensed code) must carry the same licence as the original code. According to the plaintiffs, by training their AI systems on public repositories, the defendants have violated the rights of many developers who posted their code under different open-source licences that require attribution. The claim of the plaintiffs is that Copilot neither identifies the owner of the copyright nor provides any attribution. Moreover, no Copyright Notice or any Licence Terms are attached to the output. In a nutshell, the plaintiffs pleaded for:

- Violation of the Digital Millennium Copyright Act (“DMCA”);
- Common law breach of open-source licences;
- Common law tortious interference in a contractual relationship;
- Common law fraud;
- False designation of origin
- Unjust enrichment
- Unfair competition
- Breach of contract for violation of the GitHub Privacy Policy and Terms of Service;
- Violation of the California Consumer Privacy Act (“CCPA”);
- Common law negligence;
- Common law civil conspiracy; and
- Declaratory relief

On the other hand, GitHub hinges its argument on the fact that Copilot uses the codes in a transformative way, buttressed by the provision of fair use, not amounting to any copyright infringement. In May 2023, the US District Judge in Northern California

[199] Doe 1 et al. v. GitHub, Inc. et al.,[2022], 4:22-cv-06823-JST, U.S District Court, Northern District of California, San Francisco Division.

dismissed the pleas for a declaratory relief and common law civil conspiracy and dismissed the other claims with a leave to amend. This means that the litigation shall continue to be the basis of the revised complaint.

This case presents various legal problems for Copilot-like AI based coding tools both at the stage of Free Software licensed ingested works used for AI training as well as the output of the AI itself. Regarding the input infringement, at the outset, the plaintiffs have contested that the defendants are liable for creating a derivative work simply through the act of AI model training.

From an EU perspective, The EU Directive on Copyright and Related Rights in Digital Single Market[200], provides a limited Text and Data Mining (TDM) exception:

- Art 3, which provides mandatory exception but it only applies to research organisations or culture heritage institutions using TDM for the purpose of scientific research
- Art 4, which provides a broader exception for all entities but it only applies if "the use of works and other subject matter has not been expressly reserved by their rights holders in an appropriate manner."

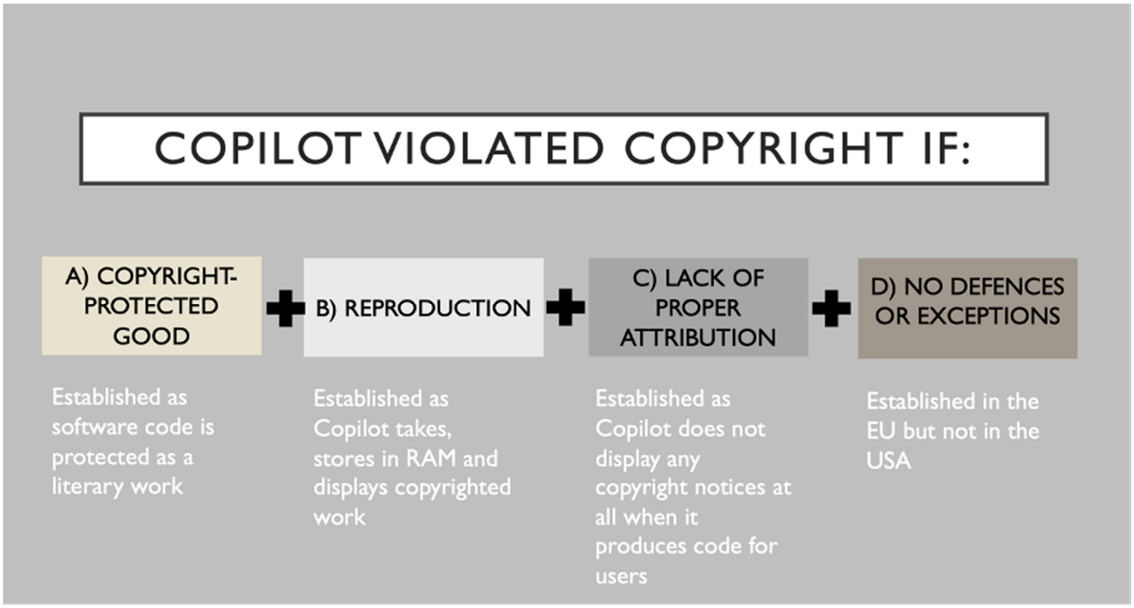
Regarding Art 4, it can be inferred that publishing code on a public GitHub repository, and consequently agreeing to the terms of service of GitHub that permit the viewing, usage, indexing, and analysis of the public code, can be interpreted as allowing mining on the published code. However, the complexity stems from the fact that there are third party codes also hosted on the same repository. Furthermore, it is unclear whether an AI training falls within the scope of this exception.[201]

As regards the output infringement is concerned, the Court observed in the GitHub Copilot case, that if the plaintiffs' code is reproduced as output, it will be reproduced in a manner that violates the open-source licences under which plaintiffs published their code. In the amended complaint, the plaintiffs have broadened the scope of these reproductions from actual reproductions, to also "functionally equivalent" codes. However, it is established that copyright protection only extends to the expression and not the functionality. Despite that, can the idea/expression dichotomy be applied to AI technologies to claim that the model only contains ideas? Assuming if the AI output

[200] Directive (EU) 2019/790 of the European Parliament and of the Council of 17 April 2019 on copyright and related rights in the Digital Single Market and amending Directives 96/9/EC and 2001/29/EC.

[201] Gabriele Montanari, 'The GitHub Copilot Case' [2022] 4iP Council <https://www.4ipcouncil.com/research/github-copilot-case> accessed 28 September 2023.

renders the original work unrecognisable and allows the author from receiving an appropriate remuneration for the use of the protected work,[202] then the Courts would need to carefully navigate through these muddy waters to render a decision. The complexities of the case as elucidated above can be summarised by way of the table below:



COPILOT VIOLATED COPYRIGHT IF:

A) COPYRIGHT-PROTECTED GOOD ✚ B) REPRODUCTION ✚ C) LACK OF PROPER ATTRIBUTION ✚ D) NO DEFENCES OR EXCEPTIONS

Established as software code is protected as a literary work

Established as Copilot takes, stores in RAM and displays copyrighted work

Established as Copilot does not display any copyright notices at all when it produces code for users

Established in the EU but not in the USA

Figure 16. Requisites for copyright violation within the GitHub Copilot case[203]

This licence compliance controversy raises many critical questions, namely as follows:

- Will the use of Copilot be considered a derivative work of the original copyleft-licensed code?
- Is there a way to identify which AI models contain derivative works?
- How can researchers use AI to train and produce outputs using publicly available datasets?

Of importance is the fact that the proliferation of AI systems is directly linked to its AI models which are vastly trained on publicly available datasets. The usage of publicly available dataset is governed by the dataset licence associated with the dataset, akin to the Free Software licences. These dataset licences outline the rights and obligations of the users and determine the context in which the datasets can be used

[202] As observed in CJEU 3 June 2021, CV-Online Latvia / Melons SIA, C-762/19, EU:C:2021:434, paragraphs 38, 46.

[203] Linda Novobilska, 'Free Open Source Licensing Requirements and Copyright Infringement Involving AI Technologies' (Masters thesis, Humboldt University 2023).

in order to ensure its legally compliant usage. That said, its compliance is not as easy as it is for Free Software licences. Free Software licences grant the use of code to be used as computer programs, however, Free Software licences are silent on use of the code as data for training AI models. Moreover, a public dataset may be hosted in multiple locations and created from multiple data sources, each of which may have different licences and thus it is difficult to locate and identify the correct dataset licences associated with a given dataset.[204] These ambiguities have highlighted some important legal debates in the GitHub Copilot case.

4.5.2. AI and copyright lawsuits

In the field of generative AI, another major lawsuit has caused a major stir in the AI and copyright domain. In Jan 2023, a class action suit[205] was filed in San Francisco, CA against three companies, Stability AI, Midjourney and DeviantArt, Inc on behalf of artists whose works were used to train AI algorithms. The complaint delves into the technology behind Stable Diffusion which relies on a mathematical process called diffusion to store compressed copies of training images, which are recombined to generate new images. The artists contend that Stable Diffusion has generated copyrighted images without their knowledge or consent. The major rebuttal of defendants is that the copyrighted data is covered by the doctrine of fair use, at least in the USA. However, the legal implications are different indifferent jurisdictions and the use of the images through data scraping is yet to be settled in the court. The defendants have also challenged the contentions of the artists on technical counts submitting that AI art models do not store images at all, but rather mathematical representations of patterns collected from these images.

Joining the barrage of lawsuits against Stability AI's Stable Diffusion, is another one filed by Getty Images[206] on the ground of their images being scrapped by Stability AI for training Stable Diffusion and AI image generation, which amounts to a derivative work. Further allegations include Stability AI removing or altering Getty Images' copyright management information, providing false copyright management information, and infringing Getty Images' famous trademarks.

[204] Gopi Krishnan Rajbahadur and others, 'Can I Use This Publicly Available Dataset to Build Commercial AI Software? -- A Case Study on Publicly Available Image Datasets' (arXiv, 11 April 2022) <http://arxiv.org/abs/2111.02374> accessed 2 November 2022.
[205] Andersen v. Stability AI Ltd, [2023], 3:23-cv-00201, U.S. District Court for the Northern District of California.
[206] Getty Images (US) Inc v. Stability AI Inc, [2023], 1:23-cv-00135 ,U.S. District Court for the District of Delaware.

The above-mentioned lawsuits demonstrate the risks involved in non-compliance with Free Software licences. There is an imminent need to train the AI technologies to respect the Free Software licence by providing correct attribution and complying with the licence terms. The disregard of Free Software licences is unfair not only to the artists and their creative works but also to the Free Software community in general which aims to foster open innovation.

In relation to copyright royalties, companies like Shutterstock are aiming to offer economic trade-offs to artists, photographers, and designers through “Contributor Funds” which will be used to pay them when the content they uploaded is sold by to firms like OpenAI in order to develop generative AI models.[207] However, the outcome of the same is questionable since at the outset, there is still an absence of consent taken from the artists to contribute to the development of generative AI models. It is also unknown as to how artists contributing to the development of the generative AI models can be identified to seek consent or to even compensate. As regards the compensation itself is concerned, there is neither a guideline in place which directs a fair compensation nor an auditing framework, rendering the practice unethical and inequitable.[208] Furthermore, this approach may not be adopted across the board by all companies to compensate the aggrieved artists/authors and is also not in alignment with the fundamentals of Free Software i.e. freedom to use, study, share and improve, as discussed in detail in the previous section.

Similar initiatives have also been made in this space. For example, Deviant Art developed a metatag that disallows generative AI models to scrape images of the artists who have prohibited their works from being used for the development of generative AI models. The ramification of this metatag is that third parties using DeviantArt-sourced content for AI training will have to ensure that their data sets exclude content that has the tags present.[209] In a similar attempt, Mat Dryhurst and Holly Herndon have developed a standard, Source+ which provides an opt in or opt out mechanism to not just artists but also musicians and writers as well to allow their work to be used as training data for AI.[210]

[207] ‘SHUTTERSTOCK PARTNERS WITH OPENAI AND LEADS THE WAY TO BRING AI-GENERATED CONTENT TO ALL’ <https://www.shutterstock.com/press/20435> accessed 28 September 2023.

[208] Natasha Lomas, ‘Shutterstock to Integrate OpenAI’s DALL-E 2 and Launch Fund for Contributor Artists’ <https://ca.news.yahoo.com/shutterstock-integrate-openais-dall-e-120126511.html> accessed 28 September 2023.

[209] See more at: https://www.deviantart.com/team/journal/A-New-Standard-for-Opting-Out-of-AI-Datasets-934500371

[210] See more at: ttps://www.inverse.com/input/culture/mat-dryhurst-holly-herndon-artists-ai-spawning-source-dall-e-midjourney.

4.6. Hybrid nature of AI

Understanding the hybrid nature of AI is essential to be able to assess the openness of AI and ultimately to define, what truly open source AI means, and how it can be built.

The main phases of the process of building AI starts with the training data that is used to train the untrained model, ending up in a trained model that is ready to be applied to input data in order to derive to the output. These phases, on such a general level, do not yet reveal sufficient details. One step towards a more detailed view, is to take a look at components involved in these phases.

Before one has the training data available to train the untrained model, original data needs to be prepared, one needs to have access to it, it needs to be collected, pre-processed and labelled. This in itself is not an easy task. In addition to working with the data, one needs to build the untrained model, its components and structure, and optimise it. But one also needs something else than merely the model and data, one might use predefined models and knowledge representations, to name some of the other components.

In the training phase, there are additional components involved. As a result of the AI pipeline, we get the trained model, but we also get or use parameters, weights, topology, activation functions, and representations or re-representations.

And finally, looking at the last part of the phases, when the output is reached. One receives an answer to the question one is asking or another kind of classification data, or one receives prediction data, for instance in the form of code or an image.

All these phases and components form a process flow that could be visualised in the following manner (Figure 17).

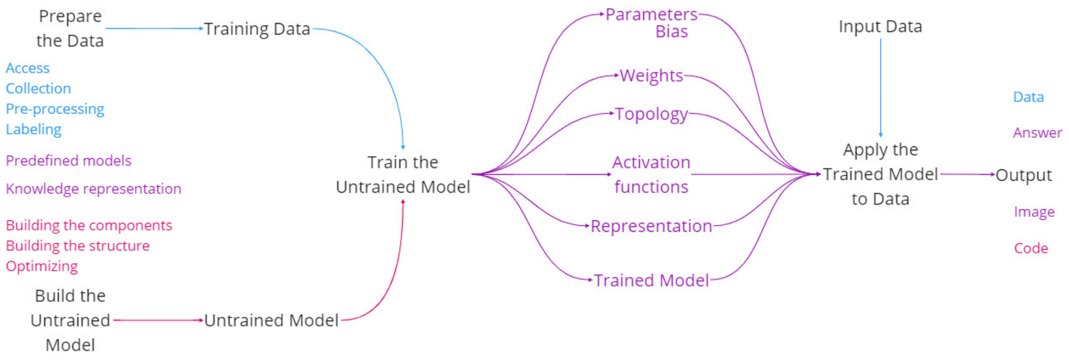


Figure 17. Phases and components of building AI

Analysing the phases and components from the perspective of different forms of property, one can identify layers of intellectual property. In the case of software, one can identify the copyright regime, and in the case of data, the layer refers to digital property or commodity of a different kind, see in more detail Subsection 3.3.3. above.

In the following visualisation (*Figure 18*) components in the sphere of data have been highlighted on a blue background as a top layer, and components falling under the sphere of copyright protected software in red. However, for several components, it is difficult to assign a clear property group. Things like parameters and weights are quite close to data, but not quite. And the model, containing or expressed in code, falls close to software. In addition to these, a considerable number of components exist, for which it is not easy to identify a traditional category of property.

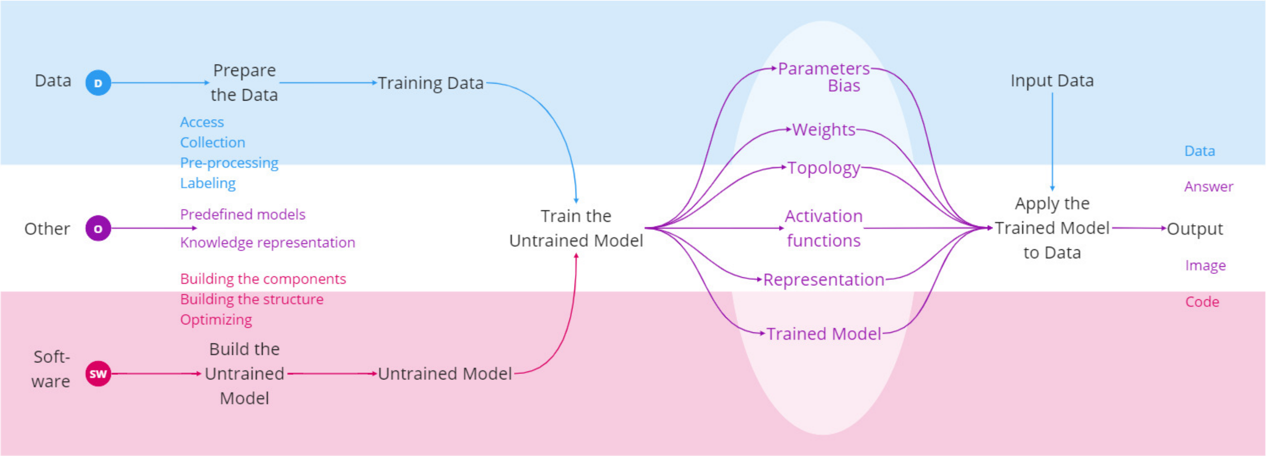


Figure 18. Property layers of AI

4.7. Towards a definition of Open source AI

Mechanisms built for open source software through principles and rules embodied in different FOSS licence terms have led to the establishment of communities, nourishing environment for co-operation and, ultimately, to flourishing industry and business. When defining open source AI, the core question is, how can similar dynamics be built into it as we have for open source software? This would require the integration of the FOSS dynamics into a very different set of hybrid assets, as shown above.

The key for the open source AI definition is in understanding how the components generated and used in different phases of building AI can be fed back to the system, i.e., to the cycle of preparing the data, building and training the models and applying the trained model to further data. As some of these components are not readable to humans, human-computer interaction is one core aspect to be taken into account in building a functioning system for open source AI. Communication between machine made ingredients and human interpretations of such ingredients is vital. Such human-computer interaction should focus on points that are essential to subsequent tasks in the pipeline of open source AI development, and leave out parts that are not necessary to the next phases of the development.

Three core requirements emerge when striving towards this kind of dynamics in open source AI. These are transparency, enablement and reproducibility (*Figure 19*).

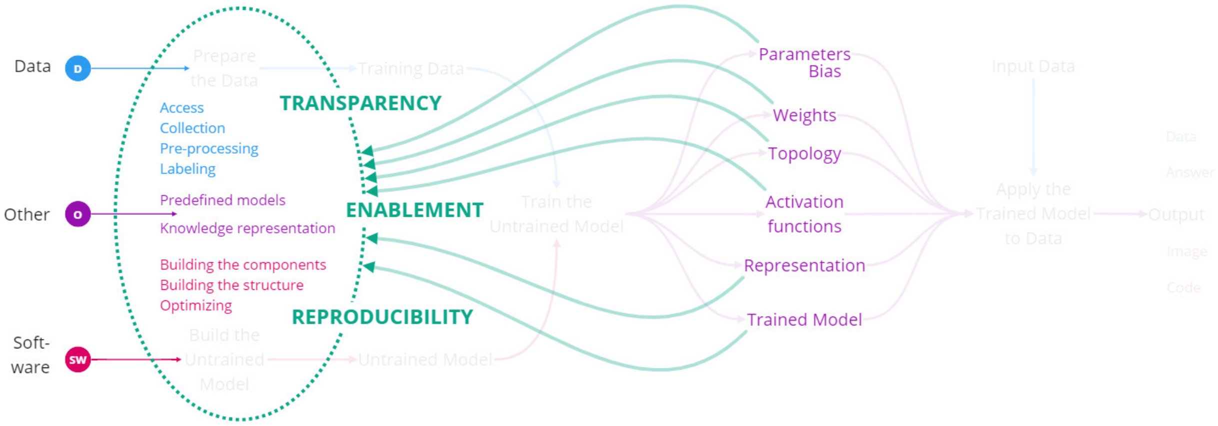


Figure 19. Requirements for building open source AI

Transparency means the disclosure of details, eg, the composition of training data sets, details about the data structures, architecture, algorithms, and access to neural network weights. Transparency in the technical sense of the word implies such level of disclosure that is conducive to enable the community around an open AI project to rebuild the model, if it so wishes.

Enablement is a principle akin to what we know from patent law. This is meant to fulfil sufficiency of disclosure in a way that is sufficient in detail, so that the person skilled in the art could carry out the acclaimed invention. Understood in this way, sufficient details about the building of a model would require, for example, model cards (see Subsection X above) or other types of disclosure and collaboration policies. Because of the hybrid nature of AI, this is going to be much broader than what exists for FOSS.

Reproducibility, means verifying the plausibility of the statement by being able to empirically track it back to the source, meaning for instance reproducible builds. This would require software development practices that create an independently variable path from the source to the binary code. Furthermore, reproducibility is related to the availability of infrastructure in the form of clusters of tensor processors which can be prohibitively expensive. Open source AI should therefore focus on the so-called ‘GPU poor’. Indeed, it is precisely community projects and collaboration that have enabled AI inference on resource-constrained devices (eg, llama.cpp, which allows running the LLaMA model using 4-bit integer quantisation on an Apple MacBook). Reducing the entry barriers to open source AI collaboration should also improve the ability to reproduce models which may otherwise be a prohibitively expensive or unreasonable activity (eg, due to environmental concerns).

At the core of traditional open source software licensing lies the possibility to run and rebuild from source (freedom 0), to modify (freedom 1) and to redistribute modifications (freedom 3). Open source AI providing the means for rebuilding through transparency, enablement and reproducibility is yet to be developed. In this task the communities around FOSS and OD play a crucial role.

In addition to the above core requirements driving towards openness of AI, how much attention should be given to the use restrictions set out in several AI licences (*Figure 20*)?

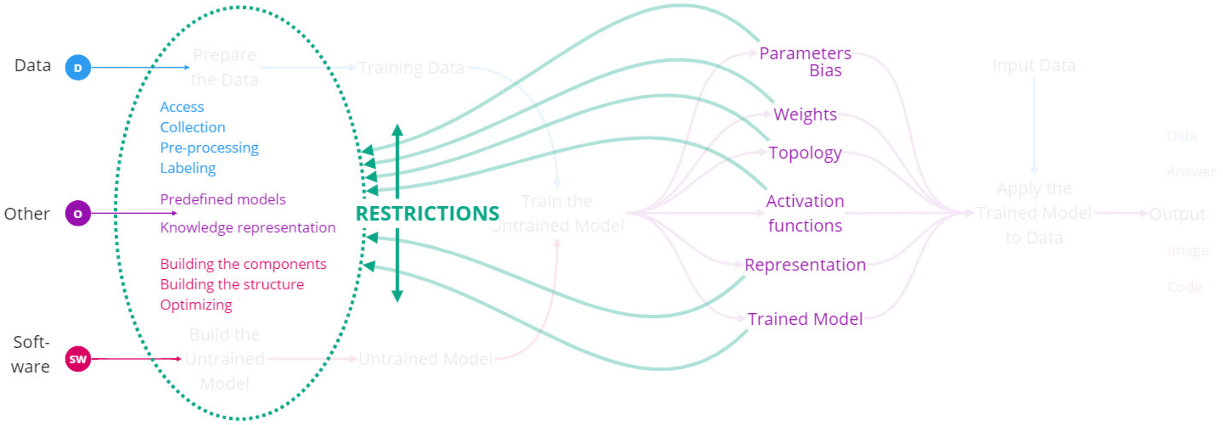


Figure 20. Role of the restrictions

In striving for responsible AI development, use restrictions have emerged in licences. This trend is fortified by the requirements from the regulators, as all over the world AI has been or will be regulated. For instance, according to the Stanford Research on Foundation Models[211], no foundation model provider seems to achieve compliance with the draft EU AI Act[212]. However, it is noteworthy that the BLOOM (Hugging Face) – focusing on the use restrictions through RAIL-licence – comes the

[211] Rishi Bommasani and others, 'Do Foundation Model Providers Comply with the EU AI Act?' (2023) <https://crfm.stanford.edu/2023/06/15/eu-ai-act.html>.
[212] Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS 2021

closest to fulfilling the requirements of the EU AI Act. When defining open source AI, the role of the regulators and aims towards responsible AI cannot be forgotten.

The use restrictions set out for instance in the RAIL-licences[213] restrict certain types of behavioural-use by the licensee, and thereby disallow certain types of applications. They also contain a copyleft-type of effect, by requiring the downstream users to respect the same restrictions. These restrictions may contain significant restrictions within the purpose, e.g., in some only academic or research purposes are allowed. Additionally, these restrictions may contain other behavioural-use restrictions aiming for responsible use, e.g., forbidding harming people, exploiting vulnerabilities, identifying machine-generated content, restricting fully automated decision making.

Such restrictions seemingly violate the Open Source Definition by OSI[214], which requires no discrimination against any person or group of persons and no discrimination against fields of endeavour, such as limiting the use to research purposes. On the other hand, the licences with behavioural-use restrictions actually aim for instance for non-discrimination by setting such restrictions. How could this kind of paradox be unravelled?

The key to addressing this paradox lies in understanding the difference in the nature of software and data, both in their forms of legal protection and in their technical nature. Software is protected by copyright, whereas data is not. Data may be protected through European database directive[215], or it may have protection through trade-secret protection regime, or its protection may simply rely on contractual practices. What comes to the difference in their technical nature, source code of the software is human-readable. Source code reveals the idea and workings behind the executable code. Similar instruments are not directly applicable to data. As data exists in different forms (e.g., raw data, annotated data, datasets, real-time data and data products), data needs different types of interpretation before it is human-readable, i.e. in the form of information or knowledge. Only after such interpretation, data is usable by different groups of people.

[213] Danish Contractor and others, 'From RAIL to Open RAIL: Topologies of RAIL Licenses' (*Responsible AI Licenses (RAIL)*, 18 August 2022) <https://www.licenses.ai/blog/2022/8/18/naming-convention-of-responsible-ai-licenses> accessed 28 September 2023.
[214] Open Source Initiative (n 3).
[215] Consolidated text: Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases

If we look at the AI licence scene we can identify, e.g., the following licence terms, some of which have been analysed in more detail above, and specify their core focus:

Montreal data licence tool (2019)[216]:

Montreal data licence identifies a taxonomy for licensing of data in AI and ML. It focuses on data and the relationship between data and the models. It is not based on openness as such, but gives some foundations from the perspective of data.

Big Science BLOOM RAIL licence 1.0 (2022)[217]:

Big Science BLOOM RAIL licence is a model specific licence, focusing on restrictions and aiming to keep an open and permissive character while striving for responsible use of the model.

Open weights permissive licence (2023)[218]:

Open weights permissive licence gives a definition and licence terms for open source compatible licensing of neural network weights (NNWs). It underlines the differences between open source software and NNWs with focus on openness. Thereby, it reveals the need to focus on the human-machine interaction perspective, an aspect that is so fundamentally different in AI compared to source code.

OpenRAIL-DAMS licence family (2022)[219]:

OpenRAIL-DAMS is a family of licences aiming to prevent irresponsible and harmful applications and, thus, focuses on the restrictions. However, it also highlights differences and combinations of data, applications/executables, models and source code.

All of the above licences make important observations on the hybrid nature of AI. By combining these observations to the dynamics that the FOSS-licences create, we are able to assess the building of open source AI from different perspectives (Figure 21).

[216] Benjamin and others (n 120).

[217] 'License - a Hugging Face Space by Bigscience' <https://huggingface.co/spaces/bigscience/license> accessed 24 September 2023.

[218] mergenthaler, 'Definition/Open Weights License.MD' <https://github.com/Open-Weights/Definition/blob/main/Open%20Weights%20License.MD>.

[219] Responsible AI (n 177).

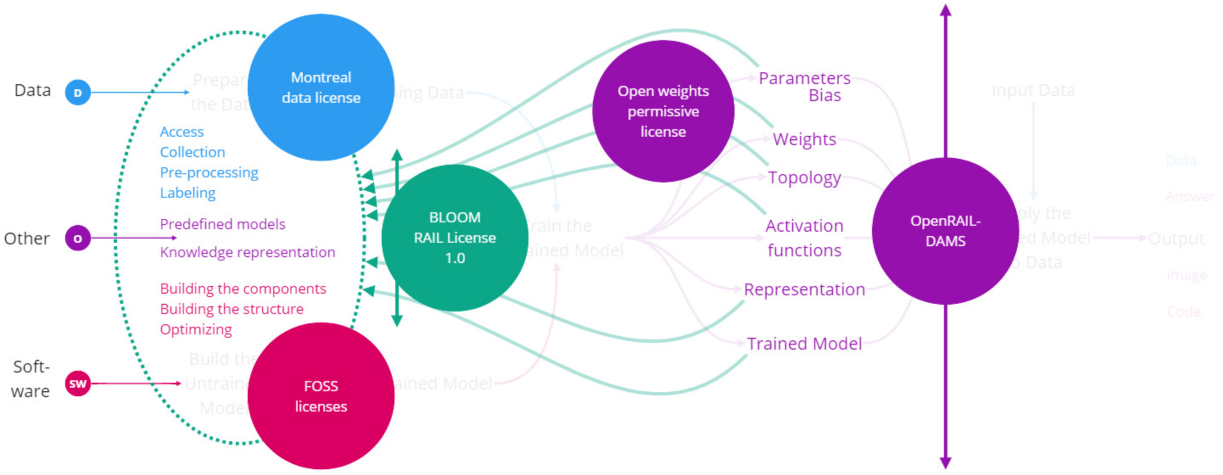


Figure 21. Perspectives to building open source AI

4.8. Conclusion

We have primarily examined the interaction of Free Software with AI technologies and thereby the challenges posed by its use on three major counts — transparency, restrictive licences used to implement “ethical” usage of AI technologies and the current litigation regarding copyright violations caused by AI. Some controversial cases worldwide were contextualised on how the black-box phenomenon projected by AI systems leads to lack of transparency in the society causing grave human rights violation. In light of that, Free Software was presented as a helpful tool for users to control, monitor and correct AI systems. Initiatives like Public Money, Public Code have been pushed forward to promote digital environments of openness and transparency through the implementation of Free Software.

A critical aspect of the paper was to analyse the regulatory framework for AI in the EU in order to highlight the challenges posed to Free Software, and how rules on liability of the software can affect developers and enterprises working and commercialising Free Software. The strategy adopted by the FSFE for different legislative processes involving the AI Act were presented and contextualised over a larger picture to demonstrate how different rules can affect the AI environment.

The work also focused on the challenges at the intersection of Free Software and AI. We highlighted how the rights rachet model can be counterproductive for Free Software as the model poses severe limitations over how the reuse of the source code affects the "four freedoms" of software. We then elaborated on the ethical dimensions of the convergence of AI and Free Software, examining the various restrictive licences designed from an "ethical" standpoint that supposedly call themselves "open" but are in fact in contravention of the definitions of Free Software and Open Source Software. We brought forth the various inadequacies in the application of these restrictive licences and questioned the very purpose and existence of the same. We concluded that given the proliferation of these licences with behavioural restrictions, integration of Free Software, licence compatibility and licensing compliance become more complex, affecting the whole ecosystem. There is a strong need for the companies adopting these licences to re-visit their Free Software licensing framework and align with the principles of the Free Software licences. The issue becomes even more critical where the public money is being used to fund the projects which at some stage exhibit restrictive practices.

In the last part of the chapter, we underlined the problem of the use of publicly available Free Software licensed codes for training AI systems particularly in light of the GitHub Copilot case. We raised many open-ended questions such as — how can researchers use AI to train and produce outputs using publicly available datasets? — as brought to light through this licence compliance controversy that need to be settled by the legislators in order to propel the use of Free Software in AI technologies. In line with the discussions on the challenges posed by AI, we briefly delved into the major lawsuits filed on the ground of copyright violations caused by AI systems, namely the various lawsuits concerning the Stability AI case. In that last section, we highlighted the way forward in relation to countering some of the above-mentioned problems. We presented the various initiatives proposed by different companies albeit being potentially contentious, such as offering economic trade-offs to contributors through a Contributor Fund, opt-in/ opt-out mechanism and metatags for development of generative AI. To sum up the discussion, the Free Software and AI can offer tremendous benefits for society if companies truly embrace the Free Software principles and avoid imposing restrictive conditions in a bid to foster ethical and responsible AI advancements. Governments should, in this regard, be particularly vary of these developments especially where public funding is involved as that hinders openness, transparency and collaborative efforts for innovation in the society.

5. Propositions for the implementation of the ZOOOM licensing framework

The specific implementation of the ZOOOM licensing framework depends on identifying the key variables in the business model, building an inventory of all intellectual property assets, and developing a sound IP strategy. The following sections provide a list of best practices for the implementation of this framework, a guide on choosing a licensing model, good practices on managing your licensing information in a standardised way, and an overview of the utility of open source program offices (OSPOs) for supporting knowledge creators.

5.1. Best practices checklist

Companies must first conduct a due diligence exercise to understand whether their business model depends on the exclusive rights of third-parties, especially patents. We can identify four main scenarios that raise IP issues:[220]

- **Use of third-party patented technology**. The technology used is covered partly or wholly by a third-party patent and the company either has no licence to practise the invention or the licence has unfavourable terms.
- **Failure to protect in-house developed new and inventive technology**. A second scenario is where the company has developed a new and inventive technology but has failed to protect it and has allowed competitors to copy the technology.
- **Use of third-party trade marks**. A third scenario is where the company has chosen a product or service name that is already protected by a registered trade mark.
- **Dependency on third-party data**. If the technology developed depends on third-party non-personal or personal data, it may be challenging or even impossible to realise the business model. This part of the due diligence analysis should combine copyright and data protection considerations.

5.1.1. Create an inventory of existing IP rights and organise a discovery session to ensure completeness

[220] Based on examples in Schriek et al., at 1-2.

To understand whether such risks and dependencies are present for the company's intended business model, it must first engage in a discovery session. This session includes discovery of third-party IP rights whose licensing is essential for the business (background IP).

We recommend using FSFE's REUSE tool for software and manual analysis for other types of subject matter. In the case of patents, we recommend retaining the services of a patent attorney who should conduct a freedom-to-operate search to establish if the intended business model impinges on third-party patent rights.

5.1.2. Check the rules on employee contributions to open source projects and employee inventions

Companies operating within the realms of open source software, open hardware, and open data must possess a keen awareness of the intricate legal frameworks governing ownership rights pertaining to innovations produced within their organisations. The genesis of intellectual creations, such as software programs, databases, and novel product designs, is not solely contingent on a company's material resources, physical infrastructure, and equipment; it equally hinges on the talent and inventive spirit of the innovators involved. Consequently, the fruits of creative endeavours within a company emerge as a product and reflection of both the employer's and the innovator's contributions. The development of new software, hardware, and data necessitates a harmonious fusion of material assets with the imaginative and intellectual prowess of the innovator. Consequently, the ownership of rights over the outcomes of innovation within companies can be ascribed to both the employers and the innovators.

Research substantiates that the majority of innovations arise within the employment context, chiefly from the efforts of employees (as referenced in Jaworski 2006, p. 125; Colston and Galloway 2010, p. 77). Nevertheless, endeavours tied to open source software, open hardware, and open data introduce a unique dimension. These companies, operating within the open source paradigm, engage not only their employees but also external collaborators, including students and independent contractors. Furthermore, these 'external collaborators' may hail from distinct legal jurisdictions, a characteristic particularly prevalent in open innovation scenarios that frequently involve online collaboration among creators from diverse countries and legal frameworks.

The issue of intellectual property (IP) rights ownership straddles the intersection of intellectual property law and labour law. Labour law fundamentally posits that the outcomes of an employee's labour belong to the employer, whereas intellectual property law confers exclusive rights over intellectual accomplishments to their creators, whether workers or independent contractors. National legislative models governing employers' rights to intellectual products vary considerably from one country to another (for a comprehensive overview, see Wolk and Szkalej 2018). The European Union has endeavoured to harmonise these regulations since the 1980s, achieving some level of cohesion in areas like computer programs and product designs but encountering complexities in the realm of patents.

The ramifications of proprietary rights in open source software, open hardware, and open data projects hold significance not only for companies engaged in open innovation but also for individuals navigating the intersection of employment and entrepreneurial aspirations.

Employees' copyright

The question of ownership over software and databases is harmonised with the EU Computer Programs Directive.[221] Article 2 (1)sets out that the author of a computer programme is the natural person or group of natural persons who have created the programme, or, whether the legislation of Member State permits, the legal person designated as the rightsholder. Article 2 (2) it follows that where a computer programme is created by a group of natural persons working together, , the exclusive rights will be owned jointly. Importantly, Article 2(3) contains a mandatory requirement on employees' programmes and therefore harmonises the conditions for exercising the economic rights in programs created by employees. It provides that where a computer program is created by an employee in the execution of their duties or following the instructions given by their employer, the employer exclusively shall be entitled to exercise all economic rights in the program so created, unless otherwise provided by contract. This means an automatic legal transfer of the copyright in the computer programmes, to the employers. However, if the parties agree, the employed author of a computer programme can recover the rights through a specific clause in the employment contract or a separate agreement on the exploitation of the computer programme made by the employee.

All EU member states have included in their national copyright law provisions implementing the mandatory requirement on employees' computer programmes contained in article 2 (3) of the EU Computer Programme Directive (Wolk and Szkalej 2018, 59).

In 2021, the UK's Intellectual Property Enterprise Court looked at the question of whether an employee could claim copyright over software developed at home and outside working hours.[222] The Court found that the copyright belonged to the employer (MD5), despite the fact that part of the work may have been conducted

[221] Directive 2009/24/EC.

[222] [2021] EWHC 293 (IPEC).

from the claimant's home, using personal equipment and outside normal working hours. The case concerned software for use in forensics computing, which is also the subject of the claimant's employment agreement with MD5. In the court's view, this was a strong and primary indication that it was work done in the course of his employment and it could be displaced by arguments that part of the work was done at home, regardless of the exact proportion.[223]

Since open innovation projects often involve external collaborators who are not employed by the company, it is crucial to highlight that existing legislation does not address the ownership of intellectual property rights associated with their contributions. Therefore, companies must exercise particular diligence in such instances and establish clear contractual agreements to preempt potential disputes and uncertainties concerning the ownership of IP rights.

Employees' databases

The legal regulation of ownership of employees' databases is however not harmonised on the EU level. A similar provision to Article 2 (3) of the EU Computer Programmes Directive was included in the first draft of the Directive 96/9/EC on the Legal Protection of Databases. However, it was deleted from the final version of the Directive on the Legal Protection of Databases. Recital 29 of that Directive only states that nothing prevents Member States from stipulating in national laws that where a database is created by an employee in the execution of their duties or following the instructions given by his employer, the employer shall exclusively be entitled to exercise the rights in the database so created, unless otherwise provided by contract.

Similar to what we wrote regarding the involvement of external collaborators in open software projects, it is crucial for the companies to cover the issue of ownership of databases with clear contractual agreements.

Employees designs

A design is the ornamental or aesthetic aspect of an article. Designs are applied to a great variety of products from different industries, such as machines, tools, computers etc.They can also be applied to hardware. The question of ownership over designs is unified with the EU Regulation 6/2002/EC on Community Designs.The Regulation is binding and directly applicable in all Member States. Article 14 (1) of the Regulation stipulates that the right to a Community Design shall be vested in the designer (or the designer's successor in title, and article 14 (3) provides that all design rights are automatically vested in the employer where the design is developed in the execution of the employee's duties or where the employee is following instructions given by the employer. The provision mirrors the rule in the EU Computer Programme Directive.

[223] [2021] EWHC 293 (IPEC), para 66.

However, this rule in article 14 (3) is subject to a pre-existing agreement, and gives priority to any national legal rules.

Notably, the Court of Justice of the European Union (CJEU) ruled in the case of FEIA v Cul de Sac (C-32/08) that there is no scope for national law to determine the ownership of an unregistered Community Design. In practice, owing to the unequivocal provisions regarding employees' Community unregistered designs and the absence of national provisions to the contrary, the transfer of a Community design should be upheld, even when national design regulations cover the same ground (Wolk and Szkalej 2018, p. 60).

Furthermore, given that designs can also be eligible for copyright protection, national copyright provisions may concurrently apply in situations covered by the EU Design Regulation.

Employees' inventions

Employer rights over employee inventions are regulated by national laws. At present, all EU Member States have national provisions regarding inventions developed by employees in the course of their employment. However, there is a wide diversity in those rules among countries.

Suggestions for companies

Companies that operate in open software, open hardware, and open data projects should approach the question of ownership over employees' inventions with careful consideration and clear policies. Here are some key steps and considerations they should keep in mind:

- Review existing employment agreements and agreements with external collaborators: these agreements often include clauses related to intellectual property (IP) ownership. Ensure that these agreements are clear and align with your company's approach to open projects. For individuals who are not employees, ensure that they have clear agreements in place that specify the ownership for their contributions to the project.
- Define Company Policies: clearly define your company's policies regarding IP ownership in the context of open projects. Consider whether your employees' inventions related to these projects will be treated differently from inventions created in other contexts.
- Consult Legal Experts: seek legal counsel to ensure that your company's policies and agreements comply with relevant laws and regulations in your jurisdiction. Intellectual property laws can vary widely between countries.

- Record Keeping: maintain thorough records of all contributions made by employees to open projects. This includes code commits, documentation, and any other relevant materials. This can be crucial in case of disputes or clarification of ownership.

5.1.3. Consider escrow services for software that may be an implementation of a patent-pending invention

In some jurisdictions, it is possible to deposit your software with a body, eg, a notary public or a private body and get a proof of the existence of the software and the date it was first deposited. This is obviously not needed for open source software, but may be useful in cases where a piece of software is being developed in the early stages before setting up a company, especially in cases where the software may be an implementation of an invention for which a patent application is pending.

In France, for example, the Agence de Protection des Programmes (APP)[224] is a non-governmental association that represents and defends the interests of authors of software and offers a software escrow service. It has sworn agents authorised to take the oath by orders of the Ministry of Culture who are competent to establish any infringement of the rights of authors, and in particular software authors or database producers.

Consider alternative business models

Business models evolve over time, so it is good practice to consider alternatives from the outset. This would help a company consider other assets that may need to be protected in case it has to implement an alternative business model.[225]

The company should also consider whether the same business model would be adopted for all countries and regions where the company will operate.[226] This is crucial because IP rights are territorial and the freedom-to-operate analysis may show that the business model would not be viable in all countries, eg, because of third-party rights.

Identify protectable assets in the foreground IP

Identification of the protectable subject matter in the foreground IP is an essential step in the discovery and inventory step.

Patentable inventions typically take the form of embodied objects (eg, mechanical constructions, control circuits, data collection devices etc.) or processes and

[224] See https://www.app.asso.fr/

[225] Uwe Schriek, Manuel F Juette and Stefan Golkowsky, ‘Basics of Intellectual Property Rights with a Focus on Start-Ups’ in Stefan Golkowsky (ed), *Intellectual property strategies for start-ups: a practical guide* (Edward Elgar Publishing 2022) 11.Schriek et al., at 11.

[226] ibid.

methods (eg, manufacturing processes, control methods, methods for storage, output, representation or transport communications of data).[227]

Practical considerations that should help knowledge creators identify protectable ideas include:[228]

- Clearly identify the product's customer, purchaser or user
- If more than one relevant group of customers, purchasers or users ar relevant, identify them clearly
- Specify the benefits or advantages that these various groups of potential buyers or users could derive
- Clarify the technical measures that help achieve these benefits or advantages
- Specify if there is more than one technical measure that is requires and, if yes, whether these different measures interact with each other
- Be specific about the technical problem you are trying to solve and be prepared to demonstrate whether you are solving the problem in a new way
- Clearly explain whether the problem being solved is technical, non-technical or previously unknown
- Clarify whether the measures used to solve the problem are technical in nature
- Analyse the possibility that one or more of the technical measures can be blocked and assess whether and how this would impact the customer's use of your product
- Think of any other conceivable solutions or modifications of the technical measures that could solve the problem in a similar way.

Besides the discovery of protectable ideas that may qualify as inventions, a company also needs to identify the essential open source licensed components, ie, components without which the business would not be able to sustain or exist at all. These may include:

- Off-the-shelf software, ie distributed applications
- Languages
- Frameworks
- Libraries
- Hardware designs
- Data sets
- Databases

Identify competitors and collaborators

[227] Examples from ibid 12.
[228] ibid 15.

This step is mostly relevant in cases where the business model depends on practising a patented invention. As a rule of thumb, knowledge creators should identify the scope of the substantive protection granted to third-party inventions, which is typically done as part of the freedom-to-operate search.

For these elements of the IP portfolio which may be suitable for open source collaboration, companies should consider whether they can create a community around these elements and how to best incentivise participation (eg, in the hardware community a permissive licensing model would be preferable to a reciprocal one).

Organise internal IP workshops

Identifying the protectable assets in a company can be a daunting task. Organising internal IP workshops could help overcome this difficulty. Ideally, these workshops should be attended by an IP professional, such as an IP counsel or a patent attorney.[229] However, such a workshop could also be led by someone in a technical function who has a basic but solid understanding of the main principles of IP.

The following is a list of questions to discuss during an IP workshop which will help a company develop a better IP strategy:

- Identification and assessment of existence of any contractual overrides in the absence of IP protection. For example, when a database is not protected either by copyright or the *sui generis* database right, the owner is free to determine the contractual conditions of the use of such database; patent cross-licensing etc.
- Assessment of downstream open source licence conditions in light of the framework's business components
    - Permissive conditions - they are usually fine, regardless of the business model.
    - Reciprocal conditions - exercise caution when combining with other subject matter.
    - Compare licence conditions applicable to different subject matter - are there overlaps or incompatibility between the licence grants?
    - Identify any dual or multi licensing conditions.
- Assessment of horizontal compatibility of the input subject matter
    - Are the licences applicable to all components in the input compatible with each other?
    - If not compatible, is it possible to mitigate the incompatibility, eg, remove a non-essential component. If not possible, what would be the consequences for the business model?

[229] ibid 18–19.

- Assessment of vertical compatibility of the input licence grants and the output licence for the project
    - Do we give more rights at the output than we have in the input?
        - If yes, then choose a different output licence
        - If not, check if the chosen output licence meets the business needs of the company
    - Are there residual risks of incompatibility?
        - If yes, describe how they are managed.
- Is there a licence compliance management system in place?
    - If not, is it necessary for the company to implement one in light of its short- to mid-term objectives?
    - If yes, is it based on any recognised international standard, eg, OpenChain, or industry good practice?
- Is there any risk assessment system in place? For example, a software bill of materials (SBOM) which facilitates the implementation of legal, export and security compliance measures.
- Assessment of export control regime and compliance. The export of dual use items (software/technology, eg, sent by email or by remote access of a server)) from the EU is subject to the EU Export Control regime. In addition, releases/disclosures of software source code to a foreign national in the EU or outside, and releases/disclosures of encryption source code and technology in a foreign country to a foreign national are also governed by EU export control laws. Multi-jurisdictional businesses should be cautious of any export of such technology/software.
    - Assessment of whether the software/technology can be classified as “information security” items, i.e., controlled cryptography products.
    - Assessment of whether the software/technology is for user’s personal use.
    - Assessment of whether the software/ technology transfer is for “basic scientific research”.
    - Assessment of whether the software/ technology was released is in the “public domain”.
    - Assessment of whether the software/ technology is used “for minimum necessary information for patent applications”.

5.2. Guide to choosing a licensing model

This guide aims to provide a list of general questions that a company choosing a licensing model should at least consider in light of their underlying business model. Open source releases occur inevitably, either because companies would like to promote adoption of a specific technology or to deal with problems in an existing product (eg, commoditization, lack of profitability etc.). When a company decides to release a particular asset as open source, the question about licence choice takes centre stage. The important question, however, is not so much about the specific licence but about the business goals pursued with releasing the asset as open source. This is why this guide starts from the assumption that an elaborate business model is a prerequisite for the selection of a licensing model.

There are two ways to use the guide: one is to analyse the available IP assets and make a choice of an outbound licensing model based on it; the other is to choose a licensing model and analyse what input would be necessary to implement the model. We recommend the first approach for two reasons: (1) it promotes better alignment with the business model and (2) incentivises companies to create an inventory of all their IP assets to help them make informed decisions about the best way to monetise these assets.

5.2.1. Software intensive business model

Where the business model is based mostly on **software distribution**, the following points should be considered. This scenario is the most typical and involves user and enterprise applications and code libraries.

- **IF** only permissively licensed open source components are used as input, **THEN:**
  - Company can comfortably use the software, provided it complies with the notice and attribution requirements.
  - Company can choose any type of licence for the output.
- **IF** only reciprocally licensed open source components are used as input, **THEN**:
  - The rule of thumb is to follow the strictest / narrowest licence grant
  - Always ensure that you do not grant more rights than you have.
- **IF** permissively licenced open source components are used as input **AND** they are:

  - Combined with reciprocally licensed components, **THEN** always check whether the combination is mere aggregation of separate components or a true combination of two modules into one program. Horizontal licence compatibility is essential here because any code combined with GPL code must be released under GPL. The outbound licence must be the one that has the most conditions.
  - Combined with proprietary code, **THEN** always ensure that appropriate attribution and notice requirements have been met.
  - Combined with proprietary and reciprocal, **THEN** always analyse the degree of connectedness between the modules (mere aggregation vs combination) **AND** ensure the horizontal licence compatibility of the input licences **AND** comply with the notion and attribution requirements **AND** follow the the rule of thumb that any code combined with GPL code must be released under GPL.
- **IF** you are developing a code library, **THEN** LGPL is generally a good choice because it does not require a developer using the library to release the entire application under a strong copyleft licence.

CAUTION: Interleaving proprietary code with GPL code is almost always likely to create problems in the future, so make sure that you have implemented at least some basic compliance processes that involve record keeping.

Where the software you are developing will be provided in a **SaaS configuration**, AGPLv3 is a sound choice so long as your potential customers are comfortable with the licence conditions on source code sharing. Consider that distribution may inevitably occur at some point (eg, acquisition deal, on-premises deployment and customisation for a customer) and make sure that your software has been developed with the idea that it may have to be distributed at some point.

Where the business model is based on **promotion of open standards**, a company could release implementing code under a permissive licence in the form of libraries that could be integrated in applications. Companies do this to encourage widespread adoption of new protocols or methods instead of seeking patent protection. In many such cases, the company would release the implementing code under Apache 2.0.

Where the business model is based on **dual licensing**, a company may choose to distribute a stripped-down version for the software under a reciprocal licence and a full commercial version under a proprietary licence. Software can be licensed and distributed on different terms, so co-existence between proprietary and open source licences is possible in this configuration. Importantly, however, licences can be either proprietary or open source; any attempts to modify existing or add additional conditions is likely to face challenges with the prohibition of field of use limitations and non-discrimination under the Open Source Definition.

Where the business model is based on **disruption of competitors**, usually this is the case where a market leader company has lost its edge in sales because of the emergence of competitive products. Maintaining and supporting the product becomes prohibitively expensive and the company is concerned that a competitor's product might become a de facto standard which would require even more costly reengineering efforts. By releasing its product under a licence from the GPL family, the company both creates a free substitute to the competitor's product and prevents them from privatising the code for its own products.

The following licence selection diagram could help companies make some initial choices about the most suitable licensing model for their business model.[230] It has two axes, each representing a decision along a spectrum of choices.

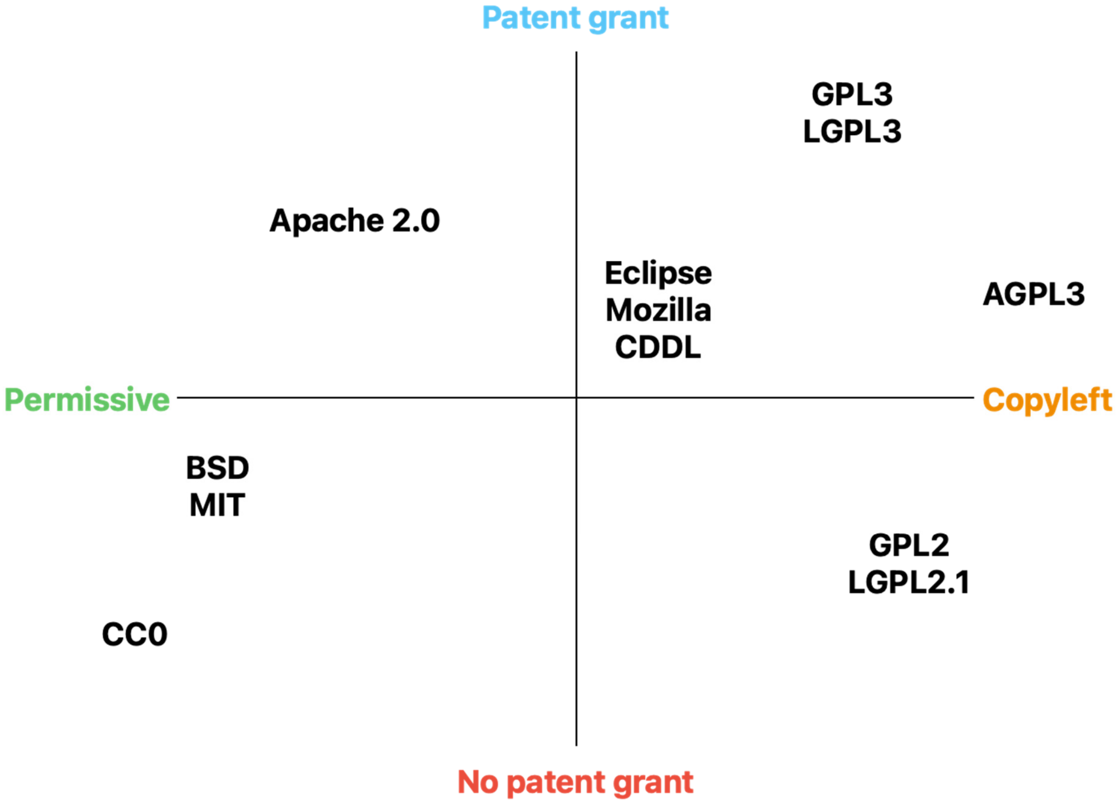


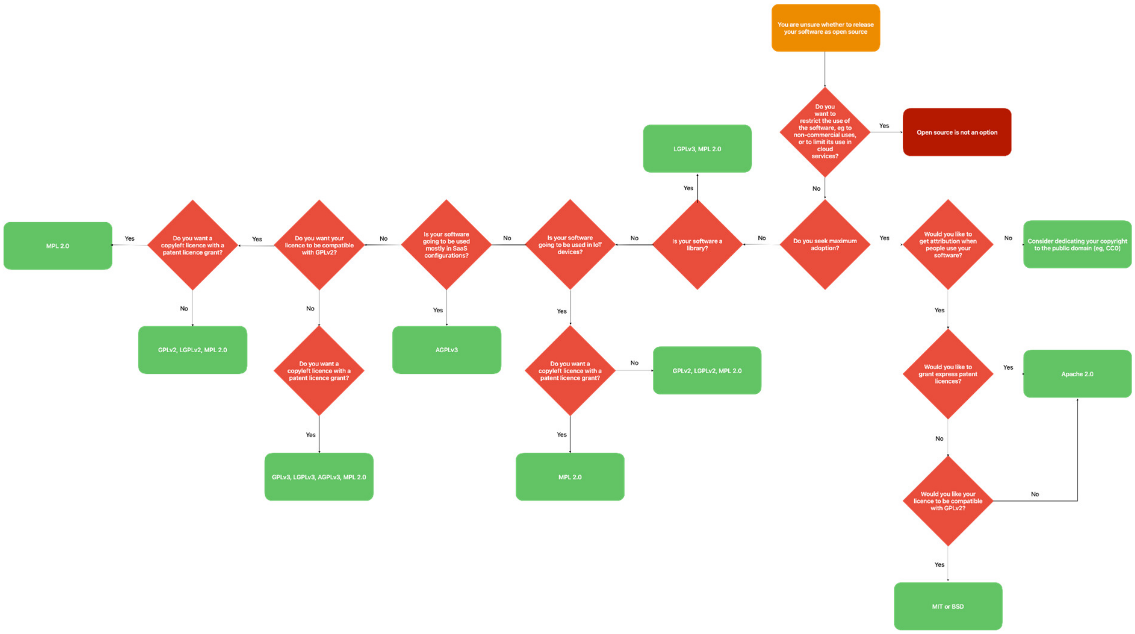
Figure 22. Licence selection diagram

[230] Adapted from Meeker (n 13) 228.

The most important factor in answering the question of which licence to choose is the choice between permissive and copyleft licences on the X-axis.

When it comes to permissive licences, the choice is not that important from a business point of view because all of them are mutually interchangeable. In fact, CC0 is a dedication to the public domain which waives all copyrights and does not require any reproduction of the licence notice. Technically speaking, CC0 it is not even a licence.

Choosing a licence involves making certain choices and compromises. The following flowchart illustrates the types of questions a company should ask and possible outcomes in terms of outbound licence.



You are unsure whether to release your software as open source
Do you want to restrict the use of the software, eg to non-commercial uses, or to limit its use in cloud services?
Yes
Open source is not an option
No
Do you seek maximum adoption?
Yes
Would you like to get attribution when people use your software?
No
Consider dedicating your copyright to the public domain (eg, CC0)
Yes
Would you like to grant express patent licences?
Yes
Apache 2.0
No
Would you like your licence to be compatible with GPLv2?
No
Yes
MIT or BSD
No
Is your software a library?
Yes
LGPLv3, MPL 2.0
No
Is your software going to be used in IoT devices?
Yes
Do you want a copyleft licence with a patent licence grant?
No
GPLv2, LGPLv2, MPL 2.0
Yes
MPL 2.0
No
Is your software going to be used mostly in SaaS configurations?
Yes
AGPLv3
No
Do you want your licence to be compatible with GPLv2?
No
Do you want a copyleft licence with a patent licence grant?
Yes
GPLv3, LGPLv3, AGPLv3, MPL 2.0
Yes
Do you want a copyleft licence with a patent licence grant?
No
GPLv2, LGPLv2, MPL 2.0
Yes
MPL 2.0

Figure 23. Licence decision process[231]

If a company decides to go for a copyleft licence, then it should make a choice between strong (eg, AGPL) and weak (eg, Mozilla) copyleft licences. Essentially, the 'strength' of the copyleft commitments is a measure of the control over how source code integrated with other code can be released.

[231] ibid 237.

The choice of whether a company would like to give an express patent grant is the key question when looking at the options on the Y-axis.

Companies that manage valuable patent portfolios are likely to prefer a licence with an express patent grant so as to avoid being caught by the uncertain scope of so-called implied patent grants. In some cases, companies might even prefer to release under a licence without a patent grant and only then grant a separate patent licence.

5.2.2. Open hardware intensive business model

Open hardware intensive business models[232] are usually organised around selling electronic components and devices (eg, Adafruit), selling name and reputation by strict monitoring of the use of a registered trade mark (eg, Arduino), selling enterprise versions (eg, SiFive), opening foundational technologies (eg, Western Digital)[233], providing open hardware consultancy and training, or selling regulatory dossiers (eg, certification in regulated industries like automotive or healthcare).

As a rule of thumb, companies should avoid licences that are not hardware-specific, such as the open source software reciprocal licences (eg, GPL, MPL) and open culture licences (CC-BY-SA). While it is in principle possible to release your designs under CC0 (equivalent to public domain dedication), unless the focus is on selling actual hardware devices coupled with a brand protection strategy (eg, through a registered trademark), this would not be most companies' primary choice. The following guidelines are much more general compared to their open source software counterparts because of the scarcity of business models, limited number of licences and the panoply of rights that extends far beyond copyright.

- **IF** the company has no interest in building a community around its designs **OR** has a business model that is organised exclusively around licensing of its patent portfolio in foundational technologies, **THEN** open hardware is not a good choice.
- **IF** the company has a strong portfolio of patents in foundational technologies **AND** would like to create a community around designs intended for release as open hardware, **THEN** a permissive open hardware licence like Solderpad v2.1 would be a good choice.

[232] Based on *Open Hardware: The Next Open Revolution? | Andrew Katz | SOOCon23 Open Hardware* (n 91).

[233] For example, the Zoned Storage concept, which is an open source, standards-based initiative to enable data centres to scale efficiently for the zettabyte storage capacity era. See https://blog.westerndigital.com/open-source-hardware-software-and-standards/

- **IF** the company has a strong portfolio of patents in foundational technologies **AND** is also interested in monetisation of an enterprise version of the project, **THEN** a mixed strategy could be adopted whereby the company releases the community edition under a permissive open hardware licence (such as Solderpad v2.1) and an enterprise edition on proprietary terms.
- **IF** the company wants to sell physical components and devices **AND** has no interest in monetisation of the underlying intellectual property rights, **THEN** it can release the project under CC-BY or any specific permissive open hardware licence. In this case, the company should have a strategy on how to establish its own brand and protect it through registered trade marks.
- **IF** the company intends to provide open hardware consultancy and services, **THEN** it can select **either** a weak copyleft (to allow for mixing with proprietary components but preserve sharing of modifications to copyleft components) or permissive licence from the CERN-OHL family, **or** a permissive open hardware licence like Solderpad v2.1, depending on the choice it wants to offer to its customers. This approach could apply also to the business case of selling regulatory dossiers for devices that require certification in regulated industries, like automotive or healthcare.
- **IF** the company wants to establish a new open specification for a foundational hardware technology, **THEN** it may consider a permissive open hardware licence that would enable it to nurture a community around the specification.

5.2.3. Data intensive business model

Before assessing the licence strategy and licence compliance issues related to open data, the company needs to assess its business strategy relating to data. The company can use the following general questions as guidance in its analysis:

- What part of the business is directly based on data and what part on other type of business, eg, other processes, products and services?
- What part of the company's activities involve open data offering and what forms the business offering of the company?
- How do the offerings in different business model categories position themselves in the added-value continuum of data, ie, starting from raw data, continuing to datasets and databases, and finishing with data products and data services?

The following guidelines need to be followed together with and in light of the IP strategy of the company and should, ideally, always be accompanied by professional legal advice:

- **IF** the company plans on using open data available from others only internally without sharing it further, **THEN** the company can use the data fairly comfortably irrespective of the category it belongs to. This may fit into open data business models aiming for cost saving, process improvement or customer loyalty. Also check the quality of the data and whether it fulfils FAIR conditions.
- **IF** the company intends to release its raw data (supporting primary business) as open data, **THEN** the company should consider using CC0-1.0 provided it fits into the company's business strategy.
- **IF** the company intends to release a database as an open asset (supporting primary business), **THEN** the company should consider which approach fits best to its business strategy: a) dedicating it to the public domain (PDDL-1.0); b) licensing it with the requirement of attribution (ODC-By-1.0); or c) licensing it with the requirement of share-alike (ODbL-1.0).
- **IF** the company intends to use open data and share such data or its derivatives, **THEN** it needs to assess to which category the open data licence belongs to:
    - **IF** the licence belongs to the category of public domain dedication (CC0-1.0, PDDL-1.0), **THEN** the use of the data and its derivatives is fairly unlimited.
    - **IF** the licence requires attribution (CC-BY-4.0, ODC-By-1.0), **THEN** remember the attribution requirements and assess whether such attribution is feasible. Note that in some cases the attribution requirements may be impractical or overwhelming.
    - **IF** the licence belongs to the group Share-Alike (CC-BY-SA.4.0, ODbL-1.0), **THEN** it can only be shared on the same licence as the original. In such cases it needs to consider its practical implications to the business strategy chosen by the company. These implications may vary depending on the type of data used, ie, whether it is open raw data, open datasets, open databases or open data products.
- **IF** the company aims to add value to the open data, open datasets and open databases, and especially when generating new data products and data services (eg, premium, freemium, dual-licensing), **THEN** the company needs to assess the underlying licences in detail and see if:
    - the restrictions set out in the licences fit the business strategy of the company;
    - the licences fit into the technology stack the company uses, and
    - there are no incompatibility issues involved regarding combinations underlying licences, note that these may arise from the fields of OD, OSS or OH.

Possible obstacles in the added-value use, include for instance overly burdensome or impossible attribution requirements, technical interoperability, lack of incentives within the stakeholder ecosystem, ecosystemic value creation aspects, and regulatory requirements.

In addition to the direct benefits that sharing of open data may bring, the companies should also consider the indirect benefits that may be gained. Sharing of open data may drive the main business of the company, for instance if the main business is in software development, platform building or in emerging hybrid technologies, like AI. Consider also indirect benefits that may be gained from participating in open data ecosystems (ODEs) or in the standardisation initiatives.

5.3. Standardising licence and compliance information

The more external components a software code encompasses, the more difficult it is for project runners to maintain a clear overview of the copyright holders and their licensing choices for each of these components. Indeed, the multiple competing requirements for communicating information about licences and copyright may increase the compliance burden on project maintainers, especially for smaller Free Software projects. Since Free Software licences are public documents being openly shared, often by millions of users worldwide, their implementation generally does not involve negotiation among the parties. Therefore, proper information regarding the governing licence is crucial to avoid legal and security risks. This can be especially problematic for Free Software projects, as large public code repositories increasingly mean licensed repositories.[234] Moreover, licence proliferation fragments the requirements for copyright and licence notices. Software projects incorporating content elements (such as text, images, and videos) face an additional layer of complexity with content licensing compliance.[235]

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

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

How copyright and licence information should be displayed in order to achieve this content licensing compliance depends on copyright law and licence requirements.[236] Especially important are notices for reciprocal licences (copyleft), as they require the derivative work to be licensed under the same licensing terms, which directly impacts licence compatibility. Although Free Software licences in general provide information on how the licence notices should be applied, the vastly diverse recommendations remain unharmonised. Standardising these licence notices in each source file is beneficial as it helps in an automated analysis of applicable licences.

This chapter will explore why standardising licence and compliance information is so important, through highlighting the importance of compliance, and the pitfalls and problems faced when this information is not standardised. It will also look at existing solutions for standardisation, including the Free Software Foundation Europe's REUSE specification.[237]

5.3.1. The multi-source model of a contemporary software project

Traditionally, platforms and software stacks were implemented using proprietary software, and consisted of various software building blocks that originated because of internal development or via third party commercial software providers with negotiated licensing terms.[238] This business environment was predictable, and companies could mitigate potential risks through licence and contract negotiations with software vendors. Under such conditions, it was relatively easy to identify the provider of every software component in the stack.

Since its inception, Free Software has expanded from an idealistic movement to build a digital commons, to a key part of the IT strategy of many organisations. As a result of these developments, many software projects, including large companies developing proprietary programmes for commercial use, have since started to incorporate Free Software components into their platforms and software stacks, resulting in the emergence of software projects that fall under the categorization of the “multi-source development” model.

[236] See Matija Suklje, ‘How and Why to Properly Write Copyright Statements in Your Code … and Probably More than What You Ever Wanted to Know about Them’ (2021) <https://matija.suklje.name/how-and-why-to-properly-write-copyright-statements-in-your-code.>.
[237] See the REUSE project's web portal, available at: https://reuse.software
[238] Haddad (n 63) 16–19.

Under this multi-source development model, digital products now have any combination of proprietary original code, third-party commercial code, and Free-Software-licensed code, all incorporated and integrated into a single software programme.

Proprietary original code

Proprietary original code is developed by the company building the software, and licensed under a software licence that does not provide the users with one or more of the Four Freedoms to use, study, share, or improve the code.

Proprietary code may be code that is developed by the company while containing some Free Software licensed code, or code that is originally developed by the company, and then integrated in Free Software components, but not contributed back to the upstream Free Software project.

Third-party commercial code

This refers to code developed by third party software providers and received under a commercial licence. Such code may be proprietary software, but also has the possibility of containing Free Software code.

Code licensed under a Free Software licence

This refers to code received under a Free Software licence, which provides all users of the code with the four freedoms of Free Software, namely the freedoms to use, study, share, and improve.[239]

Under this multi-source development model, software components consist of source code originating from any number of different sources and can be licensed under different licences. For example, a software component A can include proprietary source code in addition to third party commercial code, while software component B can include proprietary source code in addition to source code from a Free Software project.

[239] For more information on the four freedoms of Free Software, see https://fsfe.org/freesoftware/freesoftware.en.html.

5.3.2. Licence compliance in a software project

When a software project adopts a multi-source development model, they must understand the implications of having numerous different licences (and the effects of the various combinations of different licences), as well as having large numbers of licensors or contributors (copyright holders) that may number in the hundreds or even thousands (depending on the size of the project), exist within the same software project. As a result, the risk that software projects used to manage through licence agreements and negotiations with external parties and contributors, now have to be managed through Free Software compliance programs.

Free Software compliance is the process by which users and developers of Free Software observe copyright notices and satisfy licence obligations resulting from the use of such Free Software. A well planned and well executed compliance program will simultaneously ensure compliance with the terms and obligations of the applicable Free Software licences, while also helping a software project retain control over how they may wish to licence their own copyrighted code.

The more external components a software code encompasses, the more difficult it is to keep an overview of all of the relevant copyright holders and their licensing choices. Since Free Software licences are public documents being openly shared, often by millions of users worldwide, their implementation generally does not involve negotiation among the parties. Rather, the project runners choose a Free Software licence based on whether or not the bundle of terms of such licence fits their project needs. The licence terms of any given Free Software licence have to be accepted "as is", with no modifications; the predictability and certainty of the effects of the licence terms of any given Free Software licence is precisely what makes them such effective tools of the digital commons. Modifying a term of a Free Software licence results in uncertainty over the legal effects of this new licence term, and accordingly also results in a new Free Software licence. Additionally, once a Free Software project reaches a certain scale, the licences and the terms that apply to the code are extremely difficult to change. In contrast, nearly all proprietary licences are bespoke,[240] which allows developers of proprietary code to negotiate and modify the terms of their licences as and when they see fit to do so.

[240] Andrew Morin, Jennifer Urban, Piotr Silz, (2012) A Quick Guide to Software Licensing for the Scientist-Programmer. PLoS Comput Biol 8(7): e1002598. https://doi.org/10.1371/journal.pcbi.1002598, accessed 7 Aug 2023.

Therefore, properly displaying information regarding the governing licence of any particular file or piece of code is crucial to avoid legal and security risks, by alerting all users to the precise licence terms that covers the file or code, and providing certainty to what users can expect of the legal effects of such licence.

5.3.3. Non-compliance and its effects

Disregarding and failing to abide by the licence obligations placed on external code that is incorporated into a software project results in non-compliance, which we can also refer to as a failure of compliance. Such non-compliance can take many different forms, although the most common forms of non compliance are as follows:[241]

Failure to provide a proper attribution notice

Under copyright law, attribution is the acknowledgement in the use of a work of the work's author or copyright holder. In software licensing, an attribution notice is usually provided as a text file together with the Free Software component, which provides acknowledgement as supplied by the contributors of such components.

Failure to provide a proper licence notice

A licence notice is a file that includes the Free Software licence text that covers the applicable software component. The licence notice functions to inform any users who may inspect the component of exactly which licence terms are applicable. The lack of a proper licence notice may lead to confusion in downstream users as to the applicable licence terms, or mistaken assumptions that different licence terms apply to the software component.

Failure to provide a proper copyright notice

Similar to an attribution notice, a copyright notice is a more specific identifier that is placed on copies of a particular piece of work, or part of a particular piece of work, in order to provide attribution and inform the public at large of who the author or copyright holder is, and additional relevant information that can include when the

[241] Ibrahim Haddad, PhD, "*Open Source Compliance in the Enterprise*", 2nd Edition, 2018, page 16-19.

work was last modified, dates of previous versions, and contact information of the copyright holder.

Failure to provide a proper modification notice

A modification notice is a notification that alerts users to any modifications made to a piece of external code or component that has been incorporated into the software project. Modification notices can be made in a change log file, such as those required under the GNU General Public License (GPL) family of licences. The lack of a modification notice in a modified software component will result in confusion in downstream users as to what the exact content of the work that was created by the attributed copyright holder.

Failure to provide the source code

Some licences include in their terms a requirement to make the source code and modifications in the software project available to the public. For instance, this is one of the requirements of the GPL family of licences.

These are examples of common mistakes that software project runners make that result in non-compliance, although the list above is non-exhaustive. Non-compliance can come about in any number of ways depending on what kind of terms are present in the applicable Free Software licence that is used to cover the relevant software component in question. Nevertheless, as we can see from the list above, the most important mistakes that developers and software project runners need to look out for that result in non-compliance are related to failures to respect the copyright holder’s rights in relation to the credit of their authorship and their choice of licence obligations.

The consequences of non-compliance can be very serious, depending on the licence terms that are not being complied with. In some cases, non-compliance can lead to a situation of copyright infringement, which in certain jurisdictions can even carry criminal sanctions if such non-compliance is wilful and on a large enough scale[242]. Nevertheless, less serious cases of non-compliance amounting to copyright infringement can still lead to the non-compliant project being liable to pay damages,

[242] For instance, Article 61 of the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPs) requires that signatory countries establish criminal sanctions in the case of “willful trademark counterfeiting or copyright piracy on a commercial scale”.

or be the subject of an injunction (an order to stop using the code)[243]. In almost all cases, failure to comply with Free Software licence obligations has also resulted in public embarrassment, negative press, and damaged relations with the Free Software community[244].

Additionally, every failure to comply with Free Software licence terms further erodes software freedom and the integrity of digital technology; compliance is therefore necessary to ensure fair and equal rights that allows both competition in business and an equal playing field for hobbyist developers[245].

5.3.4. Copyright and licence notices

As important as compliance with Free Software licence terms is, it is impossible to comply with these terms, especially the notice requirements mentioned before, if a particular software project is unaware of what licences are contained within its code. In other words, if a project has reused external code, or external software components into the software programme without being aware of what licence terms cover these external components, it is almost a certainty that the project will be non-compliant in some way. Indeed, notice and attribution requirements constitute one of the simplest, but yet most important part of Free Software licensing[246]. These attributions can be dispensed with through the use of copyright and licence notices in relevant source code files in a given software program.

A copyright notice indicates to all users the copyright holder of a particular piece of work (in this case, of a piece of source code), and should also indicate the year in which the release of such work was completed. Similarly, a licence notice indicates the applicable licence and the copying permission for the particular file. All Free Software licences require that the downstream user produces a licensing notice[247].

On its face, these requirements for notices may sound simple to comply with. Nevertheless, as software becomes more advanced, its inner workings becoming more complicated, it is now extremely common for many software projects to incorporate large numbers of external components into their repositories. For many

[243] Meeker (n 13) 11.
[244] Haddad (n 63) 29.
[245] Software Freedom Conservancy, *VMWare Suit Concludes in* Germany, https://sfconservancy.org/news/2019/apr/02/vmware-no-appeal/.
[246] For more information, see Chapter 7 “Notice Requirements” in Meeker (n 13) 107–115.
[247] For more information, see “*How to Use GNU Licenses for Your Own Software*”, at https://www.gnu.org/licenses/gpl-howto.en.html#copyright-notice

independent Free Software developers, this complicates compliance, especially when these licensed components do not display the proper notices.

A common mistake seen in many repositories is when project runners make the assumption that the licence that they have chosen to cover the original code in their software project will be the only licence that is present within their project repository.

For example, a project runner A may decide to licence their project under the GPL version 3.0, and include a LICENCE file in the root directory of the project repository. A then decides to import and include into the project repository two software components, one licensed under the MIT License and written by developer B, and the other under the Apache License 2.0 written by developer C. Both these imported components have not been modified. The files in which these components are contained within the project repository should have the relevant copyright notices, to inform any external users that B and C are the respective copyright owners, as well as the relevant licence notices, to inform external users that the applicable licence terms for the code are the MIT License and the Apache License 2.0 respectively.

If A however fails to include any proper notices in relation to the imported components, an external user who inspects the project repository may not be informed that these components are not original code written by developer A, and therefore make the assumption that the copyright holder is A, and the applicable licence to these components is the GPL version 3.0. If code from these components is then reused into another software project by an external developer D, the problem of confusion arises.

This is because D may either neglect to include copyright and licence notices as well, or may include copyright and licence notices indicating that the relevant copyright holder is (incorrectly) A, and that the applicable licence is (incorrectly) the GPL version 3.0. Any further downstream usage of that code will now have the incorrect attribution and licensing information connected to it, which contributes to an unhealthy licensing ecosystem.

Bearing these factors in mind, the importance of having a standardised format to display licence and copyright information becomes clearer. Having such a standardised format for notices reduces the workload on software developers and project runners to find a way to properly communicate their licence and copyright information, which in turn encourages such notices and results in overall better compliance rates among software projects. Additionally, as more software projects adopt a standardised format for displaying licence and copyright notices, this in turn promotes more widespread adoption.

5.3.5. Initiatives to aid compliance

Many initiatives have developed to aid software projects in their compliance efforts. While there have been many tools developed in many different areas that help with compliance, for example FOSSology[248] and the ScanCode Toolkit[249] (both of which are tools that can scan a given software repository to detect licence, copyright, and export control information), this chapter will focus only on the initiatives that aid compliance in standardising licence and copyright notices.

One such initiative is the Software Package Data Exchange (SPDX). SPDX is an open standard (or format) for communicating software bill of material information, which includes provenance, licence, security, and other related information[250]. SPDX provides a common and standardised format for organisations and communities to share important data, thereby streamlining and improving compliance, security, and dependability. The SPDX specification is recognized as the international open standard for security, licence compliance, and other software supply chain artefacts. In particular, the SPDX project assigns a shorthand to each Free Software licence, called an “SPDX License Identifier”[251]. These licence identifiers can be used in licence notices to indicate without ambiguity the exact licence that applies to a particular work. The SPDX project also maintains a database that collects the licence texts of all Free Software licences, which will list each licence’s SPDX License Identifier, the full name of the licence, the full licence text (and including some editorial basic information regarding the licence), and a permanent URL.

SPDX therefore aids in standardising licence and compliance information. In particular, the SPDX License Identifier provides a standardised manner to efficiently refer to a licence when providing licence notices, without having to reproduce the full licence. As more software projects adopt the use of the SPDX License Identifier in their licence notices, this also makes the process of making licence notices more machine readable much easier.

[248] For more information, visit the FOSSology website at https://www.fossology.org/.
[249] For more information, visit the ScanCode Toolkit website at https://www.nexb.com/scancode.
[250] For more information, visit the SPDX website at https://spdx.dev/about/.
[251] For more information, visit the SPDX website at https://spdx.org/licenses/.

One factor that greatly eases the burden of compliance is the machine readability of such licence and copyright notices. As software programs grow ever more complex, they incorporate many more individual components than ever before, and can have repositories that contain thousands upon thousands of individual source code files. This growth in volume renders entirely human-dependent compliance impractical, and software project runners have to increasingly rely on digital tooling to aid in their compliance efforts. This means that contemporary copyright and licence notices must be machine readable by tools in order to be practical for compliance efforts. Indeed, as the usage of tools (such as the aforementioned FOSSology and ScanCode) to scan software repositories for licence and copyright information increases, being able to easily allow a tool to identify a specific Free Software licence from a SPDX License Identifier greatly simplifies the compliance process.

Using the groundwork laid by SPDX, the REUSE Initiative[252] (or simply, “REUSE”) is a project that attempts to standardise licence and compliance information in Free Software project repositories by utilising the SPDX specification. Founded by the Free Software Foundation Europe in 2017, REUSE introduces a set of best practices for expressing licence and copyright information in Free Software projects in a standard, recommended format with SPDX License Identifiers, which help to make license and copyright notices more machine-readable.

REUSE defines best practices for declaring copyright and licensing in an unambiguous, human- and machine-readable manner, so that the information is preserved when the file is copied and reused by third parties. By doing so, REUSE intends to facilitate management of source codes by making licensing and copyright information more consistent in how it is added to the source code as well, allowing for automating many of the processes involved. In turn, the hope is that this would improve management policies for the digital commons, improving data and metadata communication for individuals, communities, governments, and businesses.

With the prevalence of the multi-source software model, achieving licensing compliance in a software project has never been as important in order to avoid liabilities, assure the long term health and success of a software project, and achieve a healthy licensing ecosystem among software projects in the Free Software community. Promoting a standardised format for displaying licence and copyright information in requisite notices, can in turn improve overall compliance rates among

[252] For more information, visit the REUSE project website at https://reuse.software/.

software projects, by lowering the barrier to entry for novice developers to producing notices of their own.

New specifications for standardisation continue to develop and gain traction and usage within the community, for example SPDX and REUSE, and this encourages greater compliance in software projects.

5.4. Using business profiles to address 3Os

In this section, we provide a potential method that the companies can use for addressing 3Os related business, legal, and social questions. It contains hypothetical profiles of companies and their journey based on the issues identified and discussed above in the ZOOOM Licensing framework and Extended SCP. This includes e.g., the starting point of the company, its motivations, business aims, and principles and other major questions and problems that a hypothetical company can meet in the adoption and/or use of the 3Os. The hypothetical business profiles are the following:

- **Profile 1:** *SME using FOSS for providing SAAS for performance monitoring and management*. The company provides software-as-a-service (SAAS) for performance monitoring and management to industrial production enterprises. The company has developed its own proprietary software, but is also using several open source components. The company is trying to learn the basic aspects relating to the use of FOSS in their business.
- **Profile 2:** *Company developing software for private health institutions*. The company has been using FOSS components combined with proprietary software in its products for years. Now it is considering to start contributing to FOSS and engaging more deeply with FOSS communities.
- **Profile 3:** *Start-up developing robotics tech for elderly care*. The company has an ambitious goal to disrupt the current practices in elderly care by using robotics approach. It aims to adopt open-source assets, both FOSS and OH, to maximise its growth and it is unsure how to address the regulatory requirements and what role to take in open-source communities.
- **Profile 4:** *Medium-sized company aiming to become an enabler in OD ecosystems*. The company is representative of cases where FOSS is combined with data intensive businesses. The company has several Cities (public authorities, PAs) as customers and they are interested to know how to capture value from OD.

- **Profile 5:** *Silicon chips design company with a substantive patent portfolio using and contributing to OH*. The company aims to start a sub-project based on RISC-V open source core with associated subsystem IP, tools, and software for electronic system designers (board and devkit). It seeks to create a community around the open source cores and optimise its processes. The company has a patent portfolio and is concerned that exposing it to open source collaboration may 'taint' its patents.
- **Profile 6:** *Start-up aiming to enhance sustainable agriculture with a strong 3Os community approach*. The company is drafting an ambitious strategy for the adoption of open assets (FOSS, OH and OD), and is focusing on all aspects of value creation with a prospect of becoming a focal firm of its ecosystem. Therefore it needs a deeper understanding of all aspects (business, legal, and social) relating to the 3Os.

Below is a list of eight typical variables that can affect a company’s journey towards the adoption of the 3Os. For a complete analysis with regard to each profile, see the Appendix 1 attached to this document.

- Age & Size: Organisational size ranging from start-up’s and small companies to medium- and large-sized organisations (e.g., small/new company vs. big/established company). Potential questions: Can open-source help me compete with big and established companies?

- Experience with the open assets (FOSS, OH, OD): This can range from no experience at all (but still interested) to limited or extensive experience. Potential questions: What is the role of the 3Os in our area? How to generate value from open assets?

- Technological area: This regards the industry in which the company operates and the main technological products relating to its bussiness. Potential questions: Is our activity feasible at all without reliance on open assets? What degree of knowledge does the company have as regards the market and business ecosystems that the organisation is about to enter?

- Role (user vs. maker vs. both): Companies can be users of open assets or active contributors. Depending on their choice, a decision needs to be made regarding what elements of a product will be released as open or closed (proprietary). This also includes managing the potential hybrid combinations of the open assets licensing. Potential questions: What are the main benefits of using open assets? Can my company capture value by contributing to open assets? How to meet legal sustainability when using open technologies? What kind of strategy does the organisation have related to IP management, e.g., open, closed, or mixed? Does the licence policy support the company's business aims?

- Motivations: Companies can engaged in the 3Os for both financial and non-financial reasons. Potential questions: What are the strategic advantages provided by the 3Os? What is the social impact of open assets? Is there any relevant ethical considerations to take into account?

- Attitude towards risk: Companies can have a risky vs. conservative approach when it comes to the use of open assets. Potential questions: Do benefits overcome the legal and financial risks? What kind of risk management capabilities does the company have related to the management of open licences? Can the organisation ensure licence compliance?

- Ecosystems: This regards the role of the organisation in their business and innovation ecosystem, which can vary from beginner, to active participant, to being a focal firm of an ecosystem. Potential questions include: Is our activity feasible in total autonomy? Can we interact with new types of companies and institutions?

- Communities: Organisation may have different approaches towards the community hosting the open assets of interest. Roles in community participation can vary from no experience at all to, active participation, to extensive experience in community manager. Potential questions: how can I engage more effectively with the open-source community? How can I create a community around my products? What are the advantages and disadvantages?

The above variables aim to investigate especially the structure, the business conduct, and the social conduct parts of the Extended SCP so that the company is able to assess and/or formulate the core of the ZOOOM Licensing framework, i.e., the IP strategy, the licensing strategy, the licence management and legal implications of hybrid assets, see Figure 24 below. We have decided to omit the performance related criteria from this section, as it should focus on measuring how the structure and conduct criteria have worked in action.

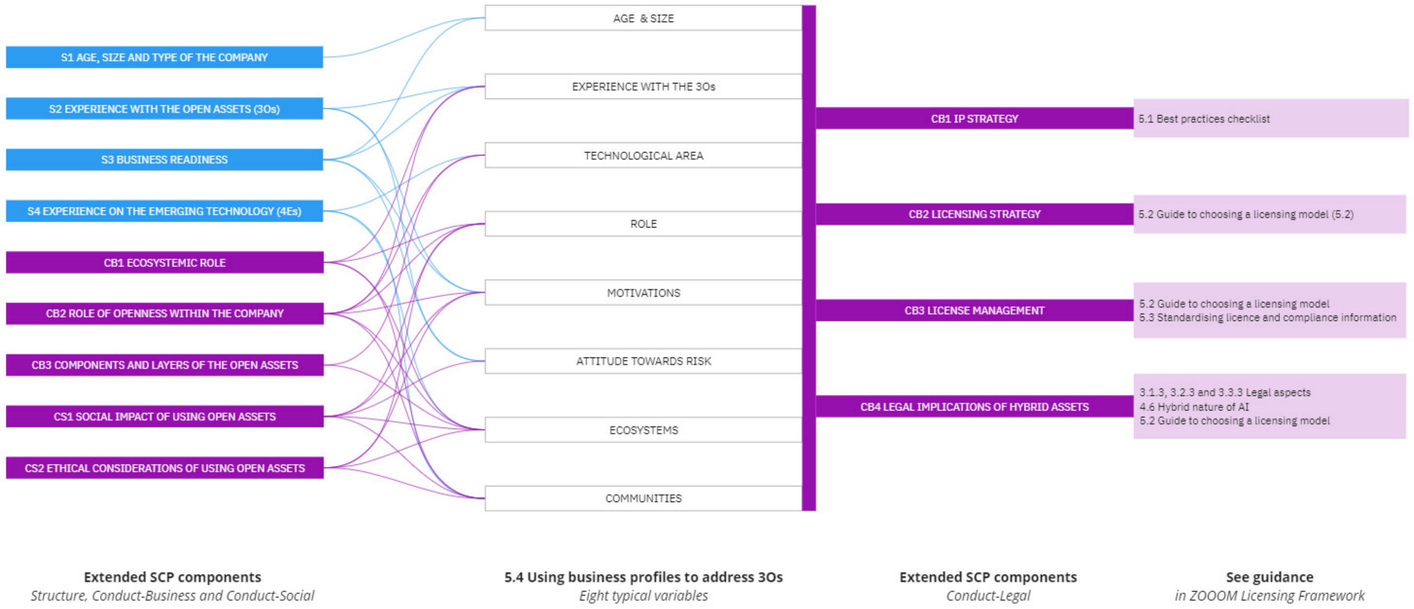

S1 AGE, SIZE AND TYPE OF THE COMPANY
S2 EXPERIENCE WITH THE OPEN ASSETS (3Os)
S3 BUSINESS READINESS
S4 EXPERIENCE ON THE EMERGING TECHNOLOGY (4Es)
CB1 ECOSYSTEMIC ROLE
CB2 ROLE OF OPENNESS WITHIN THE COMPANY
CB3 COMPONENTS AND LAYERS OF THE OPEN ASSETS
CS1 SOCIAL IMPACT OF USING OPEN ASSETS
CS2 ETHICAL CONSIDERATIONS OF USING OPEN ASSETS
AGE & SIZE
EXPERIENCE WITH THE 3Os
TECHNOLOGICAL AREA
ROLE
MOTIVATIONS
ATTITUDE TOWARDS RISK
ECOSYSTEMS
COMMUNITIES
CB1 IP STRATEGY
CB2 LICENSING STRATEGY
CB3 LICENSE MANAGEMENT
CB4 LEGAL IMPLICATIONS OF HYBRID ASSETS
5.1 Best practices checklist
5.2 Guide to choosing a licensing model (5.2)
5.2 Guide to choosing a licensing model
5.3 Standardising licence and compliance information
3.1.3, 3.2.3 and 3.3.3 Legal aspects
4.6 Hybrid nature of AI
5.2 Guide to choosing a licensing model
Extended SCP components
Structure, Conduct-Business and Conduct-Social
5.4 Using business profiles to address 3Os
Eight typical variables
Extended SCP components
Conduct-Legal
See guidance
in ZOOOM Licensing Framework

Figure 24. Interrelations between Extended SCP components, the eight variables of the business profiles and guidance in the ZOOOM licensing framework

5.5. Learnings from the Open Source Program Offices (OSPOs)

5.5.1. OSPOs in relation to the ZOOOM project scope

An open source program office (OSPO) serves as the centre of competency for an organisation's open source operations and structure, it helps to nurture, guide and align the best practises related to open source with in an organisation [253]. In the context of ZOOOM, OSPO should be considered as a model of supporting knowledge generators to correctly create, extract and manage value from the 3Os. OSPOs were born in large US companies dealing with ICT in order to share best practices and methods to interact with Open Source communities. The aim was to support the large organisations to correctly manage the IP and the interface with

[253] https://www.redhat.com/en/blog/what-does-open-source-program-office-do

open source projects from a technical department point of view. At first, they had an immediate use and an experimental character.

Over time OSPOs increased the number and type of services and differentiated based on the company business model and its complexity. These services vary from support to communities, management of licences, both inbound and outbound, through awareness raising and competence diffusion, to education, training, internal policy development in organisations, and more.

It is then reasonable to look into the various services performed by OSPOs in search of an answer to the question how to address the needs of knowledge generators in terms of advice on the use and implementation of the 3Os in the 4Es.

It is also important to note that already in recent years some supporting organisations - such as a few University TTOs and public administrations across Europe - have implemented similar models to manage the IP released in open licence projects.

The value proposition of the ZOOOM Project in relation to this model is to map and describe the possible services and best practices, guidelines and growth programmes for OSPOs.

The collected material will be organised and tailored to the needs of relevant stakeholders.

5.5.2. The history of OSPOs[254]

OSPO were designed as specific departments or branches in organisations with the role of supporting the management in the implementation and use of open source projects. OSPOs were born and conceptualised firstly in large IT companies which shaped their corporate character. OSPOs grew out of IT companies in the 80s and are linked to the open source software movement's birth and development as a reaction to the harmful effects of source code closure.

The code protected under copyleft licence in the period of time going from the 80s to the 90s saw the birth and development of communities sharing code under copyleft licences, but there was limited to no use of such code in companies. In the beginning of the 90s, some companies became interested in the open source movement and projects. In this period, companies such as IBM, Intel and others started to support various open source projects they were interested in, particularly the Linux kernel.

[254] Elaboration of the Keynote: The History and Future of OSPOs - Why they exist and role they play - https://www.youtube.com/watch?v=TAg0bKQQiok - Nithya A. Ruff, Chair, Linux Foundation Board, Head, Amazon Open Source Program Office

During the 90s, several companies contributed to the creation of foundations (eg, Apache Foundation and Linux Foundation) in order to facilitate the sharing of common and formal practices to interact with communities in a neutral and open environment so as to encourage collaboration and support development of open source projects of common interest.

It is in this context that companies such as IBM, Sun Microsystems, HP and others started to organise and create the first concepts of OSPO (Open Source Program Office) with the aim to support the different business units within their organisations in using open source software.

In this way those companies facilitated engagement with open source projects. Common characteristic of these first companies that implemented OSPOs is their core business nature in the ICT sector that was - in that period - in development both in terms of hardware architectures and software. This was driven largely by the paradigm shift from software designed for specific hardware to general interoperability of software and standard hardware architectures.

In the period from the end of the 90s and beginning of the 2000s, new types of companies were born that based their business model on internet services (.com companies) and that rapidly transformed into the well known big tech companies, such as Yahoo, Google, Facebook, Amazon etc.

At that time all architectures in use saw a strong adoption of open source software. One key change was that those companies started to contribute to open source projects that enabled their business and that created the overall online architecture. A notable example was the hyperscale technologies that were developed to implement their cloud-oriented businesses.

Those were innovative technologies - hardware and infrastructure - able to scale a computational environment in a distributed way and providing the necessary performance in terms of accessibility, fault tolerance and redundancy, typical of today’s online platforms.

Most of the topics an OSPO had to deal with were company-oriented, such as how to produce open source software, how to collaborate in communities, how to choose an outbound licence, and general best practices.

In this period, in order to standardise the approach and facilitate sharing of best practices, the TODO Group was founded. Initially, companies could only join by invitation but later access became universal and is now also extending the focus to academia and other sectors with several subtopic aims to different target groups.

Companies from outside the ICT industry are now also moving from proprietary solutions to open source projects as both users or contributors. The main driver for this switch is the necessity to swiftly adapt to the market. The approach was to move away from traditional software vendors to rapidly adapt and so limit the risk of vendor lock-in in case of variation of pricing policies, cessation of product development etc.

This drives the choice, especially for medium and large enterprises, to develop their own software and their own infrastructures also contributing to the open source projects.

In this period the number of OSPOs inside companies flourished and also integrated further functions answering to organisational objectives, from demonstrating their role in open source contribution to attracting new and qualified developers to manage and create their own projects.

In the 2000s, OSPOs adapted to the business model of organisations and their necessities. They steered enterprises towards the right use of open source software but also gradually expanded their scope to open data and open hardware.

Moreover, the idea of OSPO, originally conceived in the ICT sector, was transferred to other sectors and adopted by non-IT companies. Evidence of this is the engagement of the Linux Foundation with companies in the sectors of agriculture, energy and health.

Furthermore, the role of OSPO itself evolved to include project advising, fund raising, marketing, and stakeholder engagement.

In recent years public administrations (national, regional, cities at all levels), too, created OSPO departments to support the management and provision of software in order to increase transparency, interoperability, and avoid lock-in.

5.5.3. The role of OSPOs in knowledge valorisation

Software, hardware and data protected by IP and licensed under OSI-approved licences grant downstream users the rights to use, copy, modify and distribute said IP.

Such open IP may be used, exploited or commercialised without awareness of the terms and conditions of the licences. The scenario is also complicated by the proliferation of open licences - in particular for software, but increasingly so for data and hardware - and by the fact that new projects reuse these IP assets.

From libraries to code snippets inherited and/or refactored from other projects, datasets and less often designs, nowadays it is difficult to find an innovative product or process that will not depend on third-party IP.

As discussed, potential incompatibility between licences for the same type of assets (eg, software) and across subject matter (eg, data or hardware) may compromise non-compliant companies' business model and expose them to infringement litigation or, perhaps even worse, community derision and exclusion. It is therefore essential for organisations directly or indirectly using or creating IP under open source licences to manage this risk and to organise their activities such that they comply with the licence conditions.

Large ICT companies were the first to implement competences and to organise their operations in a way compliant with the legal requirements related to the use and creation of IP under open source licences. It is equally important for small enterprises and spin-off companies from research to get acquainted with good licensing and compliance practices.

Structures within companies or supporting organisations therefore play an essential role in guaranteeing sustainable IP management. OSPO could serve as a model to structure and organise this kind of support services and could be implemented for all 3Os both internally and externally through organisations, associations, foundations, and networks supporting IP valorisation.

5.5.4. OSPOs in the company's lifecycle

Proper IP management is a crucial process throughout the lifecycle of a company. The unlicensed use of open IP or use in breach of licence conditions could have significant legal repercussions for businesses. The following scenarios offer some examples of the role of OSPO services and their potential impact if provided externally through support organisations or internally, within the company.

Incorporation of a new innovative company

During the early stages, innovative companies operate in difficult conditions, markets have low awareness of their product, potential competitors may emerge with a similar offering, and then there are the risks of large companies exploring new markets or new technologies with larger budgets and potential interest from investors with high expectations regarding the growth of the company.

Typically, new companies are most fragile during the first three years. During this period they may be in need of support to make certain critical decisions. This is even more so because starting companies are prone to making wrong assumptions or choices regarding their business.

In this phase, IP is often the company's only asset. This means that the journey of these new companies depends on an adequate IP management strategy and valorisation processes. If the company's business model partially or wholly depends on open IP, it is also critical that the management is able to identify and present to possible stakeholders a mature and IP-aware business model.

Starting companies do not really have the financial or human resources to build a dedicated legal, compliance and community relations role at the outset of their business. Furthermore, many functions in IP management are not solely legal, compliance or organisational but are in fact a mix of these. It is therefore likely that companies would be willing to entrust IP management to an external organisation.

From this point of view, an OSPO function implemented by supporting organisations could help in setting the basis of the innovative company on the IP valorisation of the 3Os and on the identification of a sustainable business model based on that.

Business implementation

During the lifetime of the company, the business model may be enhanced, reviewed and modified in order to fit the business objectives. IP is created by innovation or development and its appropriate management guarantees a stable basis and lower risks in the activities.

It is important for the company that generates or uses IP with open licences to understand how the licensing strategies may influence decisions that are crucial for a sustainable business strategy. This aspect is relevant for inbound licences in order to be compliant and to lower the risks of licence breach that may cause legal disputes and potential business interruption. It is also relevant for outbound licensing, considering the possible issues in licence compatibility and the design of the company strategy with the 3Os.

OSPOs can supplement these competences and raise awareness on the role of open IP management. They can also support the organisations with the implementation of policy and decision making strategies to maximise the valorisation of the open IP.

Company embracing the digital transformation

The digital transformation of companies relies on the use of data to create new services with a potential high level of innovation. Being able to generate data and access data is therefore crucial for this innovation. In this sense, OSPOs could support companies in their path to digitalisation but also companies providing services, to manage effectively the data in both traditional ways, eg, trade secrets with NDAs, or sui generis database right and licensing agreements, or using open licences. Open data sets in fact may contribute to the creation of new technologies based on AI/ML which can then be implemented in new products and services.

Mergers and acquisitions

Mergers and acquisitions of companies are a natural step in the life of a company. The company’s IP is an essential asset that may influence the evaluation of the company itself. Before the merger or acquisition, consultants typically conduct a due diligence exercise to try and evaluate the IP..

Legal consultants with IP expertise are involved in the evaluation of the intangible assets. For companies with products, processes or services based wholly or partly on open IP this evaluation must pay specific attention to the business model of the company (see the consideration of business models enabled by open licences) and ensure that transfer of assets will not result in loss of rights of non-compliance.

In this example, OSPOs could facilitate such transactions. They could work in tandem with consultancy firms to perform due diligence analyses of inbound and outbound licences, build software bills of materials etc. in order to avoid that the company acquiring or the company born out of the merger could encounter legal issues post-acquisition. Furthermore, OSPOs inside the acquiring entity could support the company itself in analysing and understanding the implications of open IP for the outcome of the due diligence analysis, thereby influencing the decision on the deal’s feasibility.

5.5.5. The OSPO service portfolio

The following is a description of the services that OSPOs can implement for their stakeholders in general terms and from the perspective of currently identified best practices. In follow-up activities these services will be mapped onto the different supporting organisations in the scope of the ZOOOM Project as part of the ZOOOM Toolkit. The idea is to raise awareness about the opportunities presented by OSPOs during the second year of the Project.

Leadership and legal services

Defining legal IP strategy and policies

The goal is to support the definition of an IP management strategy for the 3Os involving different functions/departments in organisations.

- Support the definition of a strategy for reuse, contribution and valorisation of the open IP aligned with the business implementation by identifying the value created and identifying the KPIs relevant to measure it.
- Design an open source policy.

- Encourage and support management in the definition of a collaborative development strategy on identified open projects and definition of teams and budget dedicated to the contributions.
- Submit suggestions and ideas on adaptation to the business model based on the evolution of the relevant projects in 3Os
- Support in the strategic evaluation and identification of relevant projects and creation of processes to evaluate and make decisions based on the valorisation strategy, licence and risk analysis.

Due diligence for open IP

Due diligence for the use of 3Os based on the collection and analysis of the licensing models applicable to the organisation's products/processes/services .

- Raise awareness on the importance of policy creation for open IP. Organisations using 3Os could, for example, adopt policies and solutions to automate and track certain decisions on inbound licensing in the technical functions/research departments. Research institutes could define policies for the evaluation of potential impact of open IP on the specific field of research and highlight the potential for valorisation both in social and economic terms.
- General compliance monitoring to be implemented as part of the risk management and mitigation strategy in order to manage 3Os inbound and outbound licence management.
- Due diligence process should be implemented for the 3Os IP in parallel to the due diligence process on traditional IP. This process should have a strong role in verifying the licence compatibility of the different open projects that could contribute to the new IP creation and in its licensing strategy in relation to the organisation mission.
- Business models compatible and sustainable with the 3Os should be identified when considering contributing and using open IP for software, hardware and data in order to enable the innovative IP to reach the market and generate value. This aspect is surely relevant for companies using or contributing to the 3Os but should also be the basis for new companies coming from research such as new startups.

Engineering and product development support services

Bill of materials (BoM) for open software/hardware/data

The good governance of open IP starts with the collection of information of which are the actual IP assets in the organisation.

Coordination with communities

Community contribution is strategic for the creation of new projects. It is important that OSPOs could support knowledge generators in understanding community processes and adapting their approach. Moreover, OSPOs could contribute in supporting awareness raising about the new projects and therefore in soliciting participation in development activities.

One-on-one advising sessions for guidance on licensing

Collaboration on 3Os often starts from single knowledge generators that are interested in collaboration on specific projects. It is important to support these persons not only in understanding and evaluating the licences, but also in preparing feasibility studies for management bodies.

Software disclosure procedures

Software disclosure is a relevant step in valorisation of research results. This step should enable the knowledge generator to provide indications on the possible future impact on the field of research, on the societal value of the result, and on the possible opportunities for funding, visibility and investment that could be leveraged by adopting the 3Os licensing models for the IP assets.

Mapping of relevant 3Os projects

OSPOs could map and monitor projects that could be relevant to the organisation's mission with the aim of identifying the most promising ones. This could prepare the organisation to plan strategically and contribute effectively to them.

Software supply chain management services

Cyber security of the 3Os

Security, quality and vulnerability monitoring of inbound IP is a relevant issue for the security of products, particularly software. OSPOs could monitor and submit requests for updates of the 3Os projects based on software bills of materials, tracking updates and major issues arising in the inbound IP, and taking prompt action when security issues are highlighted.

Definition of contribution teams

Because of different pace and business logic among companies and communities, it may be difficult to establish a collaborative environment if the division between company development functions and employees that work on the 3Os is not clear in terms of roles, budget, time management and ownership over contributions. In this sense, OSPO could support the identification of best practices to be implemented in the organisation and suggest strategies to the management.

Communication and social engagement services

Communication and 3Os project collaboration

As previously said, projects released with 3Os licences that do not benefit from community contributions and support are going to reach only a fraction of their potential value for the overall ecosystem and the organisation. This aspect has implications on the overall quality of the project from maintenance, new developments and security.

OSPOs could play an important role of facilitating communication and collaboration by participating in sectorial events, community meetings and creating communication strategies on social media or events planning.

Community engagement

OSPOs could attract developers, other organisations or individual experts by organising events and community meetings in order to present 3Os projects, discuss future development strategies and increase the level of engagement with the community.

Liaising with external stakeholders and creating partnerships[255]

Successful 3Os projects may be willing to establish partnerships or consortia. OSPOs could support organisations in mapping and identifying these partnerships and represent the organisation in matching them with partner organisations.

Promoting industry engagement[256]

Companies and relevant stakeholders that could support 3Os projects may be incentivised to join the initiative in order to reach different markets or sectors for the project. Activities related to open innovation best practices - such as innovation challenges or hackathons on 3Os for example - could be organised or facilitated by OSPOs in order to engage the private sector in different projects and start to get to know the communities.

Human resources services

Awareness and training on the 3Os

[255] https://ospoplusplus.org/resource/trinity-college-dublin-ospo/

[256] https://ospoplusplus.org/resource/ospo-uc-santa-cruz/

Training is a crucial point of the participation of organisations in 3Os projects. Organisations need to have a common view of the basic concepts behind the 3Os across organisations.

Basics of legal compliance in 3Os

Companies need to build capacity on best practices in 3Os licensing. Training on the topic, similarly to traditional IP management, should establish common ground for organisations and, in particular, for innovative companies. Building knowledge about the compliant use of 3Os, inbound and outbound licensing, reuse of software, licence compatibility, and valorisation strategies is an essential part of the training portfolio.

The service mapping reported in this final chapter of the 5.5 section of the Licensing Framework will be the basis for the integration and the creation of services for supporting organisations and therefore will be part of the toolkit that will be created in WP3 activities.

In the WP3 there will be space to elaborate on the service map and will be structured a proposal for OSPO creation at the supporting organisation premises considering both the level of competence and expertise and the core focus of the organisation itself.

References

Arancio J and Molloy J, ‘OPEN HARDWARE IS READY TO HELP TECHNOLOGY TRANSFER OFFICES MAXIMISE THE IMPACT OF ACADEMIC RESEARCH’ (*Google Docs*, 2021) <https://drive.google.com/file/d/1E9p0_HOj9Q3QHsIr8I-NID2bYkYPxiKD/view?usp=embed_facebook> accessed 2 January 2023

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

Bego K, ‘A Vision for the Future Internet’ (Next Generation Internet 2020) Working Paper <https://www.ngi.eu/wp-content/uploads/sites/48/2020/10/Vision-for-the-future-internet-long-version-final-1.pdf>

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

Blender Foundation, ‘History’ (*blender.org*) <https://www.blender.org/about/history/> accessed 29 September 2023

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

Bommasani R and others, ‘Do Foundation Model Providers Comply with the EU AI Act?’ (2023) <https://crfm.stanford.edu/2023/06/15/eu-ai-act.html>

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

CH Open Business Event - CERN Open Hardware Licence V2 (Directed by CH Open, 2020) <https://www.youtube.com/watch?v=6wvEgQ5iWoc> accessed 29 September 2023

Contractor D and others, ‘From RAIL to Open RAIL: Topologies of RAIL Licenses’ (*Responsible AI Licenses (RAIL)*, 18 August 2022) <https://www.licenses.ai/blog/2022/8/18/naming-convention-of-responsible-ai-licenses> accessed 28 September 2023

Cowen T and Blond P, ‘"TECHNOPOLY” and What to Do about It: Reform, Redress and Regulation’ (ResPublica 2018) <https://www.respublica.org.uk/wp-content/uploads/2018/06/ResPublica-Report_Technopoly-and-what-to-do-about.pdf>

Dadgar A, ‘HashiCorp Adopts Business Source License’ (*HashiCorp Blog*, 10 August 2023) <https://www.hashicorp.com/blog/hashicorp-adopts-business-source-license> accessed 10 September 2023

Downing K, ‘AI Licensing Can’t Balance “Open” with “Responsible”’ (*Law Offices of Kate Downing*, 14 July 2023) <https://katedowninglaw.com/2023/07/13/ai-licensing-cant-balance-open-with-responsible/> accessed 28 September 2023

Eschenbach WJ von, ‘Transparency and the Black Box Problem: Why We Do Not Trust Ai’ (2021) 34 Philosophy and Technology 1607

Europe (FSFE) FSF, ‘Public Money, Public Code’ (27 September 2023) <https://publiccode.eu/en/> accessed 27 September 2023

Fogel K, ‘Delayed Open Source Publication -- Research’ (*GitLab*, 13 September 2023) <https://code.librehq.com/ots/dosp-research> accessed 23 September 2023

Free Software Foundation, ‘Frequently Asked Questions about the GNU GPL v2.0’ (*GNU Project*) <https://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.en.html> accessed 29 September 2023

——, ‘Various Licenses and Comments about Them’ (*GNU Project*) <https://www.gnu.org/licenses/license-list.html> accessed 29 September 2023

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

Free Software Foundation Europe, ‘Digital Markets Act - FSFE’ (*FSFE - Free Software Foundation Europe*) <https://fsfe.org/activities/dma/dma.html> accessed 29 September 2023

Google, ‘Patents in the Service of Open Source’ (*Google Patents Site*, 20 September 2023) <https://www.google.com/patents/opnpledge/> accessed 20 September 2023

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

——, ‘Artificial Intelligence and Data in Open Source’ (The Linux Foundation 2022) Research <https://www.linuxfoundation.org/research/artificial-intelligence-and-data-in-open-source> accessed 28 September 2023

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

Henriques-Gomes L, ‘Robodebt Class Action: Coalition Agrees to Pay $1.2bn to Settle Lawsuit’ *The Guardian* (16 November 2020) <https://www.theguardian.com/australia-news/2020/nov/16/robodebt-class-action-coalition-agrees-to-pay-12bn-to-settle-lawsuit> accessed 27 September 2023

‘High-Level Expert Group on Artificial Intelligence | Shaping Europe’s Digital Future’ (27 September 2023) <https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai> accessed 27 September 2023

Institut für Rechtsfragen der Freien und Open Source Software, ‘What Is License Compatibility?’ (*Institut für Rechtsfragen der Freien und Open Source Software*) <https://www.ifross.org/?q=en/what-license-compatibility> accessed 29 September 2023

Ito J, ‘The Issue of License Proliferation’ [2010] Joi Ito’s Web <https://joi.ito.com/weblog/2010/07/27/the-issue-of-li.html> accessed 29 September 2023

Ivo Emanuilov and others, ‘ZOOOM Deliverable 1.1 Literature Review of Legal Cases in Free and Open Source Software, Open Hardware and Open Data’ (European Commission 2023)

Javier Serrano, ‘CERN Open Hardware Licence FAQ’ (*Open Hardware Repository*, 3 August 2023) <https://ohwr.org/project/cernohl/wikis/faq> accessed 29 September 2023

Kamariotou M and Kitsios F, ‘Bringing Digital Innovation Strategies and Entrepreneurship: The Business Model Canvas in Open Data Ecosystem and Startups’ (2022) 14 Future Internet 127

Keller P and Bonato N, ‘Growth of Responsible AI Licensing. Analysis of License Use for ML Models Published on 🤗’ [2023] Open Future <https://openfuture.pubpub.org/pub/growth-of-responsible-ai-licensing/release/2> accessed 28 September 2023

Keserű J and Chan JK-S, ‘The Social Impact of Open Data.’ (Sunlight Foundation 2015) <https://www.researchgate.net/publication/298646716_The_Social_Impact_of_Open_Data#fullTextFileContent>

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

Levi S and et al, ‘Proposal for a Sovereign and Democratic Digitalisation of Europe’ (European Parliament 2021) <https://op.europa.eu/o/opportal-service/download-handler?identifier=dae77969-7812-11ec-9136-01aa75ed71a1&format=pdf&language=en&productionSystem=cellar&part=>

‘License - a Hugging Face Space by Bigscience’ <https://huggingface.co/spaces/bigscience/license> accessed 24 September 2023

Lomas N, ‘Shutterstock to Integrate OpenAI’s DALL-E 2 and Launch Fund for Contributor Artists’ <https://ca.news.yahoo.com/shutterstock-integrate-openais-dall-e-120126511.html> accessed 28 September 2023

Mattu JL Julia Angwin,Lauren Kirchner,Surya, ‘How We Analyzed the COMPAS Recidivism Algorithm’ [2023] *ProPublica* <https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm> accessed 27 September 2023

Meadows C and others, ‘Stepping up for a Truly Open Source Elasticsearch | AWS Open Source Blog’ (*AWS Open Source Blog*, 21 January 2021) <https://aws.amazon.com/blogs/opensource/stepping-up-for-a-truly-open-source-elasticsearch/> accessed 20 September 2023

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

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

mergenthaler, ‘Definition/Open Weights License.MD’ <https://github.com/Open-Weights/Definition/blob/main/Open%20Weights%20License.MD>

Miedema F, *Open Science: The Very Idea* (Springer Netherlands 2022) <https://link.springer.com/10.1007/978-94-024-2115-6> accessed 29 September 2023

Mittelstadt BD and others, ‘The Ethics of Algorithms: Mapping the Debate’ (2016) 3 Big Data & Society 2053951716679679

Mökander J and Floridi L, ‘Ethics-Based Auditing to Develop Trustworthy AI’ (2021) 31 Minds and Machines 323

Montanari G, ‘The GitHub Copilot Case’ [2022] 4iP Council <https://www.4ipcouncil.com/research/github-copilot-case> accessed 28 September 2023

Moritz M and others, ‘Value Creation in Open-Source Hardware Communities: Case Study of Open Source Ecology’, *2016 Portland International Conference on*

Management of Engineering and Technology (PICMET) (IEEE 2016) <http://ieeexplore.ieee.org/document/7806517/> accessed 17 January 2023

Morozov E and Bria F, ‘Rethinking the Smart City: Democratizing Urban Technology’ (Rosa Luxemburg Stiftung, New York Office 2018) <https://rosalux.nyc/wp-content/uploads/2021/02/RLS-NYC_smart_cities_EN.pdf> accessed 29 September 2023

Morrison R, ‘Open Source Software and Open Data: Open Licensing of Software and Data for Public Policy Analysis and for Collaborative Research — Release 03’ (12 February 2021) <https://zenodo.org/record/4537157> accessed 29 September 2023

Novobilska L, ‘Free Open Source Licensing Requirements and Copyright Infringement Involving AI Technologies’ (Masters thesis, Humboldt University 2023)

Okoli C and Nguyen J, ‘Business Models for Free and Open Source Software’ [2015] SSRN Electronic Journal <http://www.ssrn.com/abstract=2568185> accessed 14 March 2023

Open Hardware: The Next Open Revolution? | Andrew Katz | SOOCon23 Open Hardware (Directed by OpenUK, 2023) <https://www.youtube.com/watch?v=tKcedMMe750> accessed 29 September 2023

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

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

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

——, ‘Report of License Proliferation Committee and Draft FAQ’ (Open Source Initiative 2006) <https://opensource.org/proliferation-report/> accessed 29 September 2023

Open Source Software and Hardware: Community, Business Models and Licensing by Andrew Katz (Directed by Instituto de Nanosistemas UNSAM -, 2022) <https://www.youtube.com/watch?v=c-TeNLTRtAg> accessed 29 September 2023

OpenAI (2023), ‘GPT-4 Technical Report’ (OpenAI) <https://cdn.openai.com/papers/gpt-4.pdf>

Patents & Open Source: Working with 'Mixed' IP Strategies (Directed by Van Lindberg, 2020) <https://www.youtube.com/watch?v=R9UKUSj8BcQ> accessed 18 September 2023

Publications Office of the European Union., *Rethinking the Impact of Open Data: A First Step towards a European Impact Assessment for Open Data.* (Publications Office 2023) <https://data.europa.eu/doi/10.2830/911822> accessed 23 September 2023

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

'Release of Largest Trained Open-Science Multilingual Language Model Ever | CNRS' (*Press Area*, 12 July 2022) <https://www.cnrs.fr/en/press/release-largest-trained-open-science-multilingual-language-model-ever> accessed 28 September 2023

Responsible AI, 'AI Licenses' (*Responsible AI Licenses (RAIL)*) <https://www.licenses.ai/ai-licenses> accessed 24 September 2023

'Rights Ratchet' (*Meshed Insights Ltd*, 15 September 2022) <https://meshedinsights.com/tag/rights-ratchet/> accessed 27 September 2023

Schreiner M, 'Metas LLaMA v2 Could Shake up the Chatbot Business' (*THE DECODER*, 16 June 2023) <https://the-decoder.com/metas-llama-v2-could-shake-up-the-chatbot-business/> accessed 28 September 2023

Schriek U, Juette MF and Golkowsky S, 'Basics of Intellectual Property Rights with a Focus on Start-Ups' in Stefan Golkowsky (ed), *Intellectual property strategies for start-ups: a practical guide* (Edward Elgar Publishing 2022)

'SHUTTERSTOCK PARTNERS WITH OPENAI AND LEADS THE WAY TO BRING AI-GENERATED CONTENT TO ALL' <https://www.shutterstock.com/press/20435> accessed 28 September 2023

Smuha NA and others, 'How the EU Can Achieve Legally Trustworthy AI: A Response to the European Commission's Proposal for an Artificial Intelligence Act' (5 August 2021) <https://papers.ssrn.com/abstract=3899991> accessed 27 September 2023

Srnicek N and De Sutter L, *Platform Capitalism* (Polity 2017)

'Stable Diffusion Could Solve a Gap in Medical Imaging Data' (*VentureBeat*, 13 February 2023) <https://venturebeat.com/ai/stable-diffusion-could-solve-a-gap-in-medical-imaging-data/> accessed 28 September 2023

Stallman R, *Free Software, Free Society: Selected Essays of Richard M. Stallman* (Free Software Foundation) <https://www.gnu.org/philosophy/fsfs/rms-essays.pdf>

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

Szpyt K, ‘Responsible AI Licenses- a Real Alter- Native to Generally Applicable Laws?’ (2020) 1 Revista Ibérica Do Direito 178

Tarkowski A, ‘Notes on BLOOM, RAIL and Openness of AI’ [2022] Open Future <https://openfuture.pubpub.org/pub/notes-on-open-ai/release/1> accessed 27 September 2023

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

van der Broek, T.A., Rijken, M. and van Oort, S.H., ‘Towards Open Development Data : A Review of Open Development Data from a NGO Perspective’ (2012) TNO 2012 P10098 <http://resolver.tudelft.nl/uuid:c1ef3a5a-155d-4139-bb47-360a401ca339>

Vincent J, ‘OpenAI Co-Founder on Company’s Past Approach to Openly Sharing Research: “We Were Wrong”’ (*The Verge*, 15 March 2023) <https://www.theverge.com/2023/3/15/23640180/openai-gpt-4-launch-closed-research-ilya-sutskever-interview> accessed 27 September 2023

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

Workers M, ‘Microsoft and GitHub Workers Support 996.ICU’ <https://github.com/MSWorkers/support.996.ICU> accessed 28 September 2023

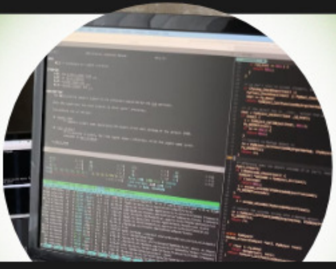
Zeleti FA and Ojo A, ‘The 6-Values Open Data Business Model Framework’ in Adegboyega Ojo and Jeremy Millard (eds), *Government 3.0 – Next Generation Government Technology Infrastructure and Services*, vol 32 (Springer International Publishing 2017) <http://link.springer.com/10.1007/978-3-319-63743-3_9> accessed 24 January 2023

Appendices

Appendix 1. ZOOOM business profiles

Profile 1

SME USING FOSS for PROVIDING SAAS FOR PERFORMANCE MONITORING AND MANAGEMENT



The company provides software-as-a-service (SAAS) for performance monitoring and management to industrial production enterprises. The company has developed its own proprietary software, but is also using several open source components. The company is trying to learn the basic aspects relating to the use of FOSS in their business.

AGE & SIZE

SME

How to ensure license compliance within the company? Should I take more structured approach to licence compliance or is ad hod decision making sufficient?

EXPERIENCE WITH THE 3Os

LIMITED

What are the core elements to understand from the FOSS licenses? Can I combine proprietary data into the FOSS components?

TECHNOLOGICAL AREA

INDUSTRIAL PERFORMANCE MANAGEMENT

How can open-source software enhance our current assets and capabilities? Are there existing open-source projects and components in this area?

ROLE

EXPERIENCED USER

Could I get more value from broadening the use of FOSS components from permissive to copyleft licenses?

MOTIVATIONS

PURELY BUSINESS STRATEGIC

What strategic advantages can I gain by using FOSS components? How can it enhance our product offerings and attract new clients? Will it increase our market position?

ATTITUDE TOWARDS RISK

RISK-TOLERANT APPROACH

What are the potential risks and challenges involved in using FOSS components?
How can I proactively mitigate these risks?

ECOSYSTEMS

EAGER TO JOIN SOME

How can I effectively integrate myself into established open-source communities and leverage their resources and expertise?

COMMUNITIES

INTERESTED BUT NOT CONTRIBUTING

Could I get more value from participating more actively in open source communities?
Does it make sense to establish collaborations within the open-source community?

Profile 2

COMPANY DEVELOPING SOFTWARE FOR PRIVATE HEALTH INSTITUTIONS

The company has been using FOSS components combined with proprietary software in its products for years. Now it is considering to start contributing to FOSS and engaging more deeply with FOSS communities.

AGE & SIZE

5-YEARS OLD (15 employees)

How can I convince my customers to change the software they have been using for five years? What is the best transition strategy?

EXPERIENCE WITH THE 3Os

NO EXP

How can I generate value from OSS? Since I need to treat data and information from patients in a secure way, is OS a possible solution or does it have critical vulnerability? Is use of FOSS always encouraged even in a heavily regulated industry like the medical industry?

TECHNOLOGICAL AREA

SOFTWARE FOR BIOMEDICAL APPLICATIONS

Can private health institutions benefit from the use of OSS instead of proprietary ones? Does my clients perceive OSS as beneficial or disadvantageous?

ROLE

MAKER & USER

Can my company capture value by contributing to OSS? How to meet legal sustainability when using OSS developed by others? Do the legal obligations differ if I am a user vis-a-vis provider?

MOTIVATIONS

COMPATIBILITY, STANDARDIZATION & SOCIAL GOOD

How to convince the user base to accept the transition? Could the adoption of OSS attract new customers, talented developers, and investors? Is there a competing product that you want to prevent from taking market share? Consider the aspect of value generation. What social good is FOSS targeting?

ATTITUDE TOWARDS RISK

MODERATE RISKY APPROACH

Could there be legal requirements stemming from regulatory and certification frameworks specific to healthcare that discourage the use of FOSS as it might be in contravention to certification and security frameworks? Is the use of FOSS advised in businesses dealing with sensitive medical data?

ECOSYSTEMS

ONLY CUSTOMER INTERACTION

Can OSS enhance interaction with non-private institutions, e.g., NGOs or the PA? Should we be concerned that a competitor might modify and use our product without 'giving back'?

COMMUNITIES

NOT ENGAGED IN ANY

How can I engage more effectively with the open-source community? How can I create a community around my products? Does it make sense at all?

Profile 3

START-UP DEVELOPING ROBOTICS TECH FOR ELDERLY CARE

The company has an ambitious goal to disrupt the current practices in elderly care by using robotics approach. It aims to adopt open-source assets, both FOSS and OH, to maximise its growth and it is unsure how to address the regulatory requirements and what role to take in open-source communities.

AGE & SIZE

START-UP

How can the company leverage open assets to start a new business?

EXPERIENCE WITH THE 3Os

NO EXP

How can I generate value from open-source assets? What are the relevant ones among OSS, OH, and OD? What are my legal rights and responsibilities if I engage in the 3Os?

TECHNOLOGICAL AREA

HEALTHCARE ROBOTICS

Can a robotics company be successful without being involved in the 3Os? How relevant are OSS, OH, and OD in this specific technological area?

ROLE

USER OR ALSO MAKER?

A company may be just a user if it relies on permissively licensed assets. In that case, the company only needs to assess legal obligations relating to the use of open assets developed by others. However, if the company relies on reciprocally licensed assets and modifies and distributes these modifications, it may be 'forced' to become a maker by the license conditions. Vetting of assets in line with the business strategy is therefore essential prior to using reciprocal open source licenses.

MOTIVATIONS

STRATEGIC

How do machine certification work with open license control software? Do you aim to get a patent over one or more inventive ideas? What are the main strategic advantages of using open assets? The company needs to identify business requirements that can leverage FOSS solutions (e.g., databases, libraries, plugins). How likely is your business model to depend on licensing royalties from proprietary technologies? Consider: exclusivity, market, territory, term, royalty

ATTITUDE TOWARDS RISK

RISKY APPROACH?

Do benefits overcome the legal and financial risks? Risks are mostly driven by the industry in which the start-up is placed. Since healthcare is a strongly regulated industry, which defines the standards to be used, the choice of how to use (and contribute to) the 3Os depends on careful assessment of those standards.

ECOSYSTEMS

NOT PART OF ANY

Is the robotics business feasible in total autonomy? The company needs to consider market maturity, market demand, regulatory frameworks, and region of operation. Can I define a certification chain to support my product?

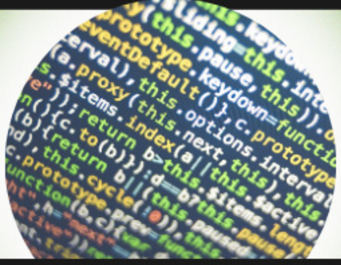
COMMUNITIES

NOT ENGAGED IN ANY

How can I create a community around my products? Does it make sense at all? It is important to acknowledge that not only business considerations determine the need community support (e.g., a community can arise around a product due to social or ethical reasons, especially in healthcare).

Profile 4

MEDIUM SIZED COMPANY AIMING TO BECOME AN ENABLER IN OPEN DATA ECOSYSTEMS



The company is representative of cases where FOSS is combined with data intensive businesses. The company has several Cities (public authorities, PAs) as customers and they are interested to know how to capture value from OD.

AGE & SIZE

MEDIUM SIZE

How can I proactively build a market for my product?
How can I create value from existing OD and new use cases combining OD to more restrictively shared data?

EXPERIENCE WITH THE 3Os

EXPERIENCED IN FOSS AND OD

If I leave my IP to the PA in the form of source code, how to make sure they will take the actions to satisfy the license? Should I convince them to contribute the IP as FOSS? What should be on my own expenses (e.g., publication and maintenance of source, forks, improvements)? To what extent should I open my own data as OD and to what extent should I keep it closed?

TECHNOLOGICAL AREA

DATA-BASED SERVICES FOR CITIES

Could I create value from data by using AI in my products?
Cities often ask for open source delivery of the IP they procured. How do I set my strategy to fit this request and to avoid loss of competitiveness?

ROLE

USER & POTENTIAL CONTRIBUTOR

How can I manage correctly the inbound and outbound licenses for sw and data? Should I use specific tools for this?
I am interested in increase data quality to produce high quality products.
Should I aim for the standardization efforts relating to data sharing? Should I take a leading role?

MOTIVATIONS

LONG-TERM AND SHORT-TERM STRATEGIES

How do I preserve my business interests and at the same time contribute to open data ecosystems? How do I combine FOSS to open data ecosystems? What kind of strategic decisions should I make?

ATTITUDE TOWARDS RISK

LOW RISK APPROACH

Copyright on sw may be the only strategical advantage I have, data is accessible to competitors. What type of business model or strategical choices can I take?

ECOSYSTEMS

PARTICIPATES IN OPEN DATA ECOSYSTEMS

How to engage with FOSS communities and to contribute effectively on open-source projects? What kind of other actors and stakeholders in the open data ecosystem? Are my competitors participating?

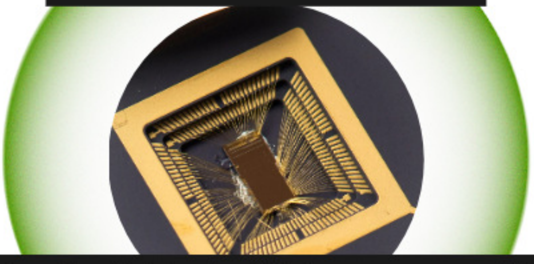
COMMUNITIES

FEW CONTRIBUTIONS ON GITHUB

Considering the required transparency (source code to the PA), how can I engage with relevant projects developers that could support me? PAs are not developers, will the code be ever used?

Profile 5

SILICON CHIPS DESIGN COMPANY WITH A SUBSTANTIVE PATENT PORTFOLIO USING AND CONTRIBUTING TO OH



The company aims to start a sub-project based on RISC-V open source core with associated subsystem IP, tools, and software for electronic system designers (board and devkit). It seeks to create a community around the open source cores and optimise its processes. The company has a patent portfolio and is concerned that exposing it to open source collaboration may 'taint' its patents.

AGE & SIZE

SME

How does the size of our company as a SME implicate the decision to initiate an open silicon sub-project? What should it consider in terms of resources, team structure, and management?

EXPERIENCE WITH THE 3Os

NO EXP

How can we measure the compatibility of our company's business model with open technologies? What challenges and opportunities does this present for starting an open silicon sub-project?

TECHNOLOGICAL AREA

OPEN SILICON

Are there essential patents of third parties that your design may necessarily infringe? How to obtain licences to potentially essential patents of competitors?

ROLE

USER or MAKER

What role should our company play in the open silicon sub-project? Are we looking to lead the project and attract contributors? How can we effectively balance our role to maximise the benefits of open source core design for efficiency?

MOTIVATIONS

STRATEGIC

Should our company become part of an open source industrial-grade ecosystem? Do we want to participate in strategic decision-making around enabling technologies? Do we want to gain industry prominence? Can we benefit from IP provenance validation as part of an ecosystem?

ATTITUDE TOWARDS RISK

WILLING TO TAKE RISKS

What potential risks and challenges should we be aware of, and how can we mitigate them effectively? Are we concerned our company IP might be 'tainted' by the rights subsisting in the new sub-project? Do we want to create a separate legal entity for this sub-project? Do we need to conduct an IP due diligence prior to launching this project within our company?

ECOSYSTEMS

NOT PART OF ANY

Are there existing open-source ecosystems, communities, or platforms that are relevant to our sub-project? How can we identify and engage with them and leverage their resources, expertise, and explore potential partnerships?

COMMUNITIES

NOT ENGAGED IN ANY

Should we adopt a policy for engagement with communities and contributor licence agreements or developer certificates of origin? How can we actively engage with the communities that align with our sub-project? What strategies should we employ to collaborate, gather feedback, and establish a positive presence within communities? How much control do we want to ascertain over the sub-project? Should it be completely community driven?

Profile 6

START-UP AIMING TO ENHANCE SUSTAINABLE AGRICULTURE WITH A STRONG 30s COMMUNITY APPROACH

The company is drafting an ambitious strategy for the adoption of open assets (FOSS, OH and OD), and is focusing on all aspects of value creation with a prospect of becoming a focal firm of its ecosystem. Therefore it needs a deeper understanding of all aspects (business, legal, and social) relating to the 30s.

AGE & SIZE

EARLY STAGE START-UP

Can open assets help me compete with bigger and more established agricultural companies?

EXPERIENCE WITH THE 30s

NO EXP

How can I generate value from open assets?

TECHNOLOGICAL AREA

PRECISION FARMING SOLUTIONS

Is sustainable agriculture feasible without reliance on open assets?

ROLE

USER or MAKER?

How to ensure legal compliance and sustainability when utilizing open technologies developed by others?
What products to release OS or proprietary? (hybrid strategy)
Consider complex legal interactions between OSS, OH and OD

MOTIVATIONS

NOT ONLY STRATEGIC

What are the main benefits of incorporating open technologies into sustainable agriculture? The company needs to make a cost-benefit analysis. How can OS provide social value to sustainable entrepreneurship?

ATTITUDE TOWARDS RISK

RISK-AVERSE APPROACH

Do the potential benefits outweigh the legal and financial risks associated with adopting open assets? The company needs to consider country- and industry-specific regulatory regulations

ECOSYSTEMS

NOT PART OF ANY

Is sustainable agriculture viable in complete isolation from open-source communities and collaborations?

COMMUNITIES

NOT ENGAGED IN ANY

How can I foster a community around my sustainable farming products? Does it make sense to establish collaborations within the open-source community?

ZOOOM

This document is an output of the ZOOOM Project

(GA 101007385)

Funded by the European Union

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