

Lukas Gabriel

Submission Date: 20/10/2023

**Open for Business: An Investigation of Open Source
Monetization and Growth Strategies**

BACHELOR'S THESIS

Submitted for the Academic Degree of

BACHELOR OF SCIENCE

First Supervisor:

Prof. Dr. Peter Weiß

Second Supervisor:

Prof. Dr.-Ing. Joachim Schuler

Statutory Declaration

I, Lukas Gabriel, hereby certify that I have written this bachelor thesis independently, that I have not used any sources or aids other than those indicated, and that I have marked all passages in the thesis that have been taken over verbatim or in spirit. Furthermore, the thesis has not yet been submitted for examination at another institution.

In addition, I assure that I have used AI-supported writing tools and that I have listed these tools completely in the Section "Overview of used tools" with their product name, my source of supply (e.g., URL) and information about the used functions of the software as well as the scope of use. In the preparation of this thesis, I have worked independently throughout and in a controlling manner in the use of AI-supported writing tools.

Stuttgart, October 20th, 2023



Overview of used tools

ChatGPT by OpenAI (paid)

- Source of supply: <https://chat.openai.com/>
- Used functions: Chat function, based on *GPT-4 & Instruct-GPT*
- Scope of use: Passive writing assistance & inspiration, assistance with programming and data visualization

Bing Chat by Microsoft (free)

- Source of supply: <https://www.bing.com/search?q=Bing+AI>
- Used functions: Chat function, based on *GPT-4 & Instruct-GPT*
- Scope of use: Literature research assistance

Acknowledgements

I would like to extend my sincerest thanks to my supervisors, Prof. Dr. Weiß and Prof. Dr.-Ing. Schuler, for their guidance and support over the last four months.

I want to thank all my colleagues, family and peers who have helped me by proofreading this thesis and inspiring me throughout.

I am also deeply grateful to those that provided their insight and experience through their case study responses.

Finally, I want to acknowledge the selfless work of the Free and Open-Source Software community, who our society is deeply indebted to.

“An open source community arises from the synchronization of the individual interests of many parties.”

- Simon Phipps, former OSI president, in:
Open Source Strategies for the Enterprise, 2012

Abstract

Open-Source Software (OSS) has played a central role in shaping the modern digital environment. Over the last five decades, many business models have evolved around value creation with OSS. However, there remains a noticeable hesitation among businesses, especially in German Small and Medium Enterprises (SMEs). This thesis offers a comprehensive overview of essential aspects of business strategy, licensing, and community management related to OSS. Utilizing a taxonomy-based approach, we examine three case studies relevant to German SMEs, aiming to derive actionable insights for businesses considering active involvement in OSS. By integrating public data with insights from participating company decision-makers, this research raises evidence in support of certain business strategies: It highlights the importance of a tailored user segmentation strategy and careful calibration of free and commercial offerings. It suggests the importance of a symbiotic community relationship and a balanced approach towards community governance. The analysis also presents evidence indicating the resilience of the Direct-sale revenue mechanism, and conversely, the possible vulnerabilities of Infrastructure-based models to intermediation. The thesis concludes by discussing current OSS trends and suggesting strategies considering trends such as licensing challenges and emerging software supply chain threats.

Keywords: open source, open-source software, business models, open-source business models, case study

Table of Contents

List of Abbreviations	VIII
Table of Figures.....	IX
List of Appendices	IX
1 Introduction	10
1.1 Research Topic and Significance	11
1.2 Research Questions and Methodology.....	12
1.3 Structure of the Thesis.....	12
2 Open Source: Definition, Characteristics, Evolution	13
2.1 A Definition of Open Source	13
2.2 Emergence and Growth of Open Source.....	15
2.3 Contemporary Open Source and Trends.....	17
2.3.1 Code Hosting and Sharing.....	18
2.3.2 Package Managers.....	19
2.3.3 Cloud-Native Technologies.....	20
2.4 Benefits of Open-Source Software	20
2.4.1 Preventing Technical Debt.....	20
2.4.2 Preventing Vendor Lock-In	21
2.4.3 The Open Source Community	22
2.4.4 Lower Costs.....	23
2.4.5 Transparency and Trust.....	23
2.5 Limitations of Open-Source Software	24
2.5.1 Lack of Support and Danger of Abandonment.....	24
2.5.2 Quality and Feature Inconsistency	25
2.5.3 Loss of Control	26
3 Open-Source Licenses.....	27
3.1 License Types.....	27
3.1.1 The Concept of Copyleft.....	28
3.1.2 Permissive Licenses	28
3.1.3 Restrictive Licenses.....	29
3.1.4 Public Domain Licenses	29
3.2 Open-Source vs. Source-Available Licensing.....	30

4	Open Source Business Models	32
4.1	Community vs. Commercial Open Source	32
4.2	Taxonomy of Open-Source Business Models.....	33
4.3	Archetypes of Open-Source Business Models.....	37
4.3.1	Infrastructure Business Model	38
4.3.2	Open-Core Platform Business Model	39
4.3.3	Professional Services Business Model.....	40
4.3.4	Open-Source Platform Business Model.....	41
4.3.5	Dual Licensing	43
5	Research Methodology	44
5.1	Multi-Case Study Design	44
5.2	Case Selection Criteria	45
5.3	Public Data Collection.....	46
5.4	Questionnaire	46
5.4.1	Selection and Participation	47
5.4.2	Privacy and Information Disclosure	47
5.4.3	Questionnaire Topics.....	48
5.4.4	Questionnaire Limitations	49
6	Case Studies and Analysis.....	50
6.1	Presentation of Individual Case Studies	50
6.2	Taxonomy Application	54
6.2.1	Value Proposition	55
6.2.2	Value Network	56
6.2.3	Value Architecture	58
6.2.4	Value Finance.....	59
6.3	Summary of Questionnaire Responses	60
7	Findings and Discussions.....	64
7.1	Key Insights Derived from Case Studies	64
7.1.1	Value Proposition	64
7.1.2	Value Network	66
7.1.3	Value Architecture	67
7.1.4	Value Finance.....	68

7.2 Research Question Discussion	69
7.3 Potential Future Challenges of Examined Companies.....	74
8 Conclusion	76
8.1 Future Developments and Strategy Recommendations	77
8.1.1 Growing Importance of Social Dynamics in Open Source	80
8.1.2 Increasing Pressure from Third-Party Service Providers	81
8.1.3 Worsening Software Supply Chain Security	83
8.2 Suggestions for Future Research	84
Bibliography.....	86
Appendix	98

List of Abbreviations

OSS	Open-Source Software
BM	Business Model
OSBM	Open Source Business Model
OSI	Open Source Initiative
OSD	Open Source Definition
FOSS	Free Open-Source Software
FLOSS	Free Libre Open-Source Software
FSF	Free Software Foundation
GPL	GNU General Public License
PR	Pull Request
CC0	Creative Commons Zero
SSPL	Server Side Public License
BSL	Business Source License
AWS	Amazon Web Services
GCP	Google Cloud Platform
GDPR	General Data Protection Regulation
IDE	Integrated Development Environment
SME	Small and Medium Enterprises
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service
UX	User Experience
VC	Venture Capital
SLA	Service Level Agreement
CLA	Contributor License Agreement
CAA	Copyright Assignment Agreement
DSA	Digital Services Act
AI	Artificial Intelligence
DMA	Digital Markets Act

Table of Figures

Fig. 1: Usage of Open Source Licenses in 2020	27
Fig. 2: Representation of the OSBM Taxonomy in a Morphological Box ..	33
Fig. 3: Placement of the Companies in Morphological Box	54

List of Appendices

Appendix 1: Code Hosting Platform Comparison Table	98
Appendix 2: Open-Source License Comparison Table.....	98
Appendix 3: Case Study Questionnaire Questions.....	101
Appendix 4: Questionnaire Response from P1 & P2.....	106
Appendix 5: Questionnaire Response from P3.....	112

1 Introduction

Originating from the foundational days of the internet, Open-Source Software (OSS)¹ has evolved to be the bedrock of today's digital world. A 2022 industry study revealed that an astonishing 97% of codebases contain OSS, with 78% of the overall lines of code derived from open-source projects.^{2,3} Within a span of just two years, the adoption rate of OSS in German companies rose from two-thirds in 2019 to three-fourths in 2021; this trend extends to the German public sector, where 64% of public authorities utilize open-source software.⁴ Conversely, only seven percent of high-ranking German IT professionals interviewed in 2021 were dismissive or sceptical of OSS.⁵

This surge in OSS adoption shows the great potential and opportunities OSS presents for organizations. More developers than ever before are actively participating in OSS: Development platform GitHub recorded a 27% growth in active users – to a total of 94 million – in 2022 alone.⁶

Whether it's transitioning existing proprietary codebases to open source or developing new projects on open-source foundations, the benefits of joining the OSS movement have never been more compelling.

Yet, despite its growing prominence, many organizations – particularly in Germany – remain hesitant to actively participate in OSS development.⁷

The intricacies of open-source dynamics, such as licensing, community stewardship, compliance, and software supply chain management, can be hard to navigate for traditional companies.⁸ These pain points are most significant in very small and very large organizations.⁹

¹ "Open Source" is only hyphenated when used as an adjective ("open-source")

² Of 2.409 codebases examined across 17 industries

³ Synopsys (2022), P. 4–10

⁴ PricewaterhouseCoopers (2021)

⁵ Bitkom (2021), P. 12

⁶ Carroll (2022)

⁷ Bitkom (2021), P. 30

⁸ Bitkom (2021), P. 14–15

⁹ Bitkom (2021), P. 12

Furthermore, a striking 72% of German companies have no strategy for OSS participation, even though a majority use and modify open-source code and recognize the opportunities of giving back to the OSS community.¹ When asked about their hesitation to make use of or contribute to OSS, IT decision-makers often mention a lack of qualified personnel, uncertainty towards legal and compliance considerations, and an arduous transition process towards open software.²

1.1 Research Topic and Significance

It is apparent that these organizations have a need for guidance in participation in OSS development. To achieve this, it's essential to analyse OSS success stories in similar companies, identify common success factors, and formulate actionable strategies for businesses willing to participate in OSS.

To rise to this challenge and take full advantage of its benefits, firms must reach a comprehensive understanding of OSS dynamics. While many studies delve into individual facets of the open-source paradigm – such as licensing, community management, or business models (BMs) – few offer a bird's-eye view of the entire path to open-source. For example, Duparc et al.³ show how OSS business models can be classified into archetypes by applying a taxonomy-based approach; Shah and Nagle⁴ achieve a comprehensive understanding of the importance of user communities; and Harutyunyan et al. identify best practices in corporate open-sourcing of existing proprietary software.⁵

This research aims to provide the aforementioned overview of the process behind OSS monetization and growth strategies. The primary objective of this thesis is to unearth the strategies and success factors that have been proven to be beneficial for businesses involved in the open-source domain.

¹ Bitkom (2021), P. 18, 22, 25

² Bitkom (2020), P. 23

³ Duparc; Möller; Jussen; et al. (2022)

⁴ Shah; Nagle (2020)

⁵ Harutyunyan; Riehle; Sathya n.d.

1.2 Research Questions and Methodology

To achieve this goal, a qualitative approach is employed, leveraging a multi-case study analysis method.

The research revolves around three research questions:

RQ1: What specific management and legal strategies have proved successful for companies producing open-source software?

RQ2: How can businesses identify and apply the most effective monetization models for their open-source projects based on their unique contexts and circumstances?

RQ3: How do open-source companies balance effective community growth and project direction while retaining control and adhering to open-source principles?

1.3 Structure of the Thesis

The structure of this thesis is designed to provide a logical and comprehensive exploration of OSS. We commence with a historical overview of OSS in Section 2.1, tracing its evolution to understand current trends and their implications on development. This leads into a discussion on the merits of OSS over proprietary software in Section 2.4, examining the motivations behind its use, development, and contribution. We also delve into reasons one might be hesitant to embrace or contribute to OSS in Section 2.5.

Subsequently, the significance of OSS licenses is highlighted in Chapter 3, with a breakdown of various license types and their business implications. In Chapter 4, we explore the Open Source Business Model (OSBM) taxonomy by Duparc et al., classifying different models into distinct archetypes.¹

¹ Duparc; Möller; Jussen; et al. (2022)

Our research methodology and data collection techniques are outlined in Chapter 5.

In Chapter 6, we present three case studies, applying the aforementioned taxonomy and employing a morphological analysis. The insights from the conducted questionnaire are showcased, followed by a synthesis of the accumulated data in Chapter 7. This synthesis aids in extracting key findings and addressing the research questions.

In Chapter 8, we conclude by projecting potential future trends in OSS, offering recommendations for businesses navigating these trends, and suggesting avenues for future research.

2 Open Source: Definition, Characteristics, Evolution

To understand the unique benefits and challenges that OSS presents for organizations, one must understand what distinguishes it from proprietary software, what the tenets of Open Source are, and how its extraordinary characteristics make it a unique paradigm in the IT industry. Furthermore, understanding how OSS first emerged can help elucidate today's OSS landscape.

2.1 A Definition of Open Source

Even though the term 'open source' is already neatly self-descriptive – referring to software whose source¹ is openly accessible by the public – it brings with it a host of other assumptions that need to be considered when discussing open source.

To answer the question “What is *open-source software*?” we can start by asking “When does *software* become *open source*?”

¹ The human-readable source code required to compile, build and run the software yourself.

Parts of a codebase will often be released to facilitate development of additions by third parties or allow independent auditing; furthermore, there have been numerous instances of entire codebases being leaked to the public.¹ This shows that the publication of the source code does not necessarily make its software open source. Instead, software becomes open source as soon as its source code is released to the public *under an open-source license* – which is how the Open Source Initiative (OSI) defines *open source*.² Therefore, one can consider OSS to be, first and foremost, a *licensing approach*. The OSI rigorously outlines this requirement in their Open Source Definition (OSD), which governs the guidelines that software – and the license under which it is released – should meet to call itself *open source*.³ Among other guarantees, the OSD mandates that OSS can be used commercially, and even sold – an important merit that enables the premise of this thesis.⁴ While the OSI is a widely recognized authority in the open source community, it cannot legally enforce the OSD or restrict the use of the term *open source*. However, it publishes a list of licenses that adhere to the OSD and pass a license review by the OSI.^{5,6}

Another topic of importance to this discussion is the subtle but very significant distinction between *open-source* software and *free* software. The term *free* in this context is not a reference to cost, but rather to the freedoms granted to its users: The freedom to run, study, share and modify the software.⁷ Since being *open source* is strongly implied for software designating itself as *free* software, the combined acronym FOSS (Free OSS) is often used; the acronym FLOSS (Free, Libre⁸ OSS) is commonly used as well, particularly to clarify the difference between the *costs of usage* and the *liberties of usage*, and place an emphasis on the latter.⁹

¹ Clement (2019)

² Open Source Initiative (2006b)

³ Open Source Initiative (2006b)

⁴ Open Source Initiative (2007), sec. 6

⁵ Open Source Initiative (2006a)

⁶ Open Source Initiative (2022)

⁷ Free Software Foundation (2023)

⁸ French & Spanish for *free* as in *freedom*

⁹ Free Software Foundation (2021)

While the term free software is championed by the Free Software Foundation (FSF), the label ‘open-source’ is promoted by the OSI. These two entities, though sharing their mission of fostering innovation and openness in the software industry, have distinct philosophies that are at odds with each other in certain areas.¹ The FSF has a strong ideological background and considers OSS as a means of social activism; it can be considered a cornerstone of the overarching Free Software Movement.² The FSF’s founder, Richard Stallman, echoes this notion and refers to the free software movement as “a movement for freedom and justice”³ The OSI takes a more pragmatic approach, focusing on the practical benefits of an open development model instead of the ethical and moral imperatives of the FSF.⁴ This has led to tensions between the FSF and the OSI, as described in Stallman’s essay “Why Open-Source Misses the Point of Free Software.”⁵

2.2 Emergence and Growth of Open Source

While the concept of *free* software has been used as early as 1980, the term *open source* only emerged in the late 1990s.⁶ In the early days of computing, software was not considered a product in its own right, but rather an inseparable part of the hardware that was shipped with it.⁷ The academic principles of sharing and openness had a profound influence in this early stage since computers were mostly operated by universities and other academic institutions.⁸ In these times, software was freely shared – between the few people that worked with computers – in a *laissez-faire* manner not too dissimilar to today’s FOSS. The construction of ARPANET in 1969 served to make the process of code sharing and collaboration even easier.⁹

¹ Free Software Foundation (2023)

² Stallman (2008)

³ Stallman (2022), P. 1

⁴ Open Source Initiative (2018)

⁵ Stallman (2022)

⁶ Open Source Initiative (2018)

⁷ Gonzalez-Barahona (2021), P. 1

⁸ DiBona; Ockman; Stone (1999), P. 1–3

⁹ von Hippel; von Krogh (2009), P. 209–210

When the software market came into existence in the 1970s – after *IBM* first started selling software independently from its hardware – proprietary software became the norm.¹ The first progenitors of *open source* as understood today were released in the early 1980s: Among them were typesetting software TeX², circuit design toolset SPICE³, and Unix⁴ – the early ancestor of today's Linux and macOS. Of these three, SPICE is of particular interest; after its release into the public domain, it quickly became the de facto standard in the electronics industry – demonstrating how a FOSS-based strategy can help a product claim a large market share.⁵ TeX and SPICE are both still actively used and developed today.

In the late 1980s, the groundwork for FOSS began taking shape with notable efforts like the GNU Project initiated by Richard Stallman in 1983 and establishment of the Free Software Foundation in 1985. This period also saw the formulation of the GNU General Public License (GPL) in 1989, a significant legal framework that used copyright law to ensure software freedom.⁶ Other prevalent OSS licenses such as the MIT⁷ and BSD⁸ licenses also emerged in this time period. By this point, FOSS communities had begun exploring various sustainability models including public funding, donations, direct company funding, and volunteer work, and had formed a legal infrastructure around GPL, BSD, and MIT license principles. This era saw the onset of digital communication within geographically dispersed FOSS communities through mailing lists, Usenet groups, and FTP servers, introducing formal organizational mechanisms like appointed GNU "maintainers" and steering committees for project stewardship.⁹

¹ Gonzalez-Barahona (2021), P. 1

² TeX Users Group (URL)

³ Engineering and Technology History Wiki (2022)

⁴ The Open Group (URL)

⁵ Gonzalez-Barahona (2021), P. 1

⁶ Rajala; Nissilä; Westerlund (2006), P. 3

⁷ FOSSA (2021c)

⁸ Haff (URL)

⁹ Gonzalez-Barahona (2021), P. 76–78

The mid-1990s witnessed the Internet evolving into a mass market service, with FOSS playing a crucial role in this expansion as it powered much of the Internet infrastructure. This era marked the beginning of businesses capitalizing on the opportunities presented by FOSS, leveraging its components to build complex systems at a fraction of the cost, and harnessing the collaborative ethos of FOSS communities for business advantage.¹ In 1998, the release of Netscape Communicator as FOSS marked a pivotal moment, leading to the coining of the term "open-source software" and the formation of the OSI, establishing a formalized approach towards open source licensing and collaboration.²

Entering the 2000s, collaborative coding platforms such as GitHub, launched in 2008, simplified collaboration and contributions to open-source projects on a global scale.³ This era also marked the increasing involvement of tech giants like Google and Microsoft in open-source initiatives.⁴ In the 2010s, the involvement of tech giants deepened, with many now leading open-source projects and actively contributing to others. For instance, Microsoft's acquisition of GitHub in 2018 signified a monumental endorsement of open source's value proposition from the corporate sector. Similarly, Google's stewardship of projects like Kubernetes exemplifies how open source is instrumental in advancing cloud technologies.

2.3 Contemporary Open Source and Trends

While the core tenets of OSS have remained the same since its emergence, its role in technology and society has changed over the years. Overarching trends such as Cloud Computing and Social Networking have influenced the way OSS is developed, distributed, and used. We quickly examine the topics of Code hosting, Package managers, and Cloud-native technologies and their impact on today's OSS landscape.

¹ Gonzalez-Barahona (2021), P. 76–78

² Aksulu; Wade (2010), P. 577

³ Carroll (2022)

⁴ Statistics and Data (URL)

2.3.1 Code Hosting and Sharing

The manner in which companies share their open-source code with the world can have a significant impact on the success of their OSS strategy. This Chapter gives a quick overview of the history of OSS code sharing, the most prevalent code hosting platforms, and their characteristics and implications thereof. A comparison table that compares these platforms in detail is provided in Appendix 1.

As described in Section 2.2, the modalities of sharing OSS code have seen significant changes over the decades. Today, integrated code-hosting platforms like GitHub and GitLab are the most popular options for personal and commercial users by a wide margin.^{1,2} These platforms offer a comprehensive suite of Git features, ranging from forks, branches, to pull requests (PRs). Moreover, they seamlessly integrate these features with essential project management and CI/CD³ tools such as Issues, Projects, and automated actions & build pipelines.⁴ Where these platforms show significant differences between one another are the areas of project management, discoverability, and social features. Each of these areas can be an immense asset to companies that aim to build a community of users and developers around their product and optimally benefit from user feedback and the talent and creativity of hobbyist developers.

For example, GitHub is the only platform that offers a ‘home feed’ which shows the user projects and releases they might be interested in.⁵ GitHub combines this with other features that one would expect from a social network – customizable profiles, achievement badges, and the ability to follow other users – to bring developers together, motivate new users to contribute,

¹ Vaughan-Nichols (2022)

² Hecht (2019)

³ Continuous Integration & Delivery: The practice of automated building, testing, and deployment of software.

⁴ Gitea Ltd. (URL)

⁵ heise online (2023)

and heighten engagement on the platform.¹ These features are complemented by a full-text global search that makes it easy to find projects – or even single lines of code – across the entire platform.² Additionally, users can “star” a code repository to bookmark it and/or show their approval. Since repositories with a fast-growing number of “stars” are promoted in users’ home feeds, this mechanism can generate organic publicity for promising new projects.³ Conversely, other platforms such as Azure DevOps and Bitbucket do not offer such features, focusing on fundamental enterprise features instead. The optimal choice of code hosting platform ultimately depends on the unique situation and objectives of the company. Companies should consider establishing a presence on another platform if its features align more closely with the goals of their OSS strategy.

2.3.2 Package Managers

Today, almost all of the most-used programming languages have an ecosystem of packages that greatly simplify dependency management and collaboration between developers.⁴ This decentralized and automated approach makes it easy for developers of any skill level to utilize the work of developers all over the world. It also plays an important role in democratizing the access to open-source code – in contrast to more centralized approaches of the 20th century, as described in Section 2.2. However, it also places a lot of trust in package maintainers, giving rise to software supply chain attacks such as the 2020 attack on the network infrastructure company SolarWinds; this attack affected thousands of organizations around the globe, including many government agencies.^{5,6} While most packages are maintained by reputable developers and organizations, it is possible for vulnerabilities or malicious code to slip through, either inadvertently or through concerted attacks.

¹ GitHub Inc. (2023b)

² GitHub Inc. (2023a)

³ Borges; Valente (2018), P. 1, 31–32

⁴ Muhammad; Real; Homer (2019), P. 1–2, 6

⁵ Martínez; Durán (2021), P. 1

⁶ U. S. Government Accountability Office (2022)

2.3.3 Cloud-Native Technologies

The commoditization of cloud services has lowered the barrier of entry to a degree where cloud servers can be provisioned for a low single-digit monthly amount, if not for free.¹ This has allowed individual consumers to take advantage of cutting-edge cloud technologies. Furthermore, containerization of applications has made it easier for developers to provide standardized execution and development environments that users and contributors can run on their machine or in the cloud. This has made using and contributing to OSS easier and more accessible – especially across different platforms, languages and ecosystems.² In the same way, the paradigm of Infrastructure as Code has made it possible to share declarative code “blueprints” of infrastructure using OSS licenses and platforms. This has placed infrastructure setup and configuration into the OSS domain, enabling developers and teams to replicate complex environments quickly and easily.³ This can introduce IT roles who might not have seen value in OSS participation – such as Infrastructure, DevOps, Network, and Site Reliability Engineers – as possible contributors to OSS.

2.4 Benefits of Open-Source Software

In assessing advantages and disadvantages, both users and developers have distinct considerations. However, many of these points apply to both sides. Hence, we now present the most important benefits and pitfalls from each side.

2.4.1 Preventing Technical Debt

When a company chooses to develop software internally, it has an obligation towards its internal users to support and maintain that software as long as it is used. Therefore, the use of open-source software which is backed by a community of developers – who themselves rely on its efficacy and are

¹ cloudcommunity (2023)

² Docker Inc. (2023)

³ Red Hat Inc. (2023b)

therefore incentivized to collaborate in its upkeep – can be an effective and straightforward means of reducing technical debt.¹

From the software maintainer’s perspective, open-sourcing can also be a means of reducing technical debt: The most common sources of technical debt in software development² are lack of documentation, insufficient testing, and non-adherence to coding best practices – all of which can be effectively addressed through open-source development.³ Including the OSS community in development provides a peer-review mechanism for the code which helps to reduce security and performance issues early on.⁴ Additionally, writing code that is known to be publicly accessible mandates developers to adhere to best practices, document their code, and thoroughly vet it for vulnerabilities.⁵ This also discourages “security by obscurity”⁶, a practice that is not recommended and is often unsuccessful at increasing the effective security of the product.^{7,8}

2.4.2 Preventing Vendor Lock-In

Proprietary software companies often employ a lock-in strategy, encouraging customers to stay within their ecosystem and maintaining high switching costs, ensuring they continue to purchase their products.⁹ Conversely, OSS developers prioritize versatility across diverse use cases and actively collaborate to bolster interoperability while minimizing lock-in. This stems from the fact that many OSS developers offer their software for free.¹⁰

¹ Sharma (2022), P. 112

² Causes related to development methodology, i.e. excluding causes related to planning and management, such as Deadlines

³ Ramač; Mandić; Taušan; et al. (2022), P. 14

⁴ Swire (2004), P. 165

⁵ Shah; Nagle (2020), P. 11

⁶ A practice that aims to increase the security of a system by hiding its specifications.

⁷ Scarfone; Jansen; Tracy (2008), P. 15

⁸ Directorate-General for Communications Networks; Blind; Pätsch; et al. (2021), P. 236

⁹ Niemi; Tuisku; Hameri; et al. (2009), P. 77–86

¹⁰ Almeida; José; José (2011), P. 2–5

Limiting interoperability in favour of business goals can risk developers forking the codebase and building in interoperability themselves.¹ This can help ensure the absence of lock-in strategies even in commercial OSS products.

2.4.3 The Open Source Community

Proprietary software is restricted by the number of developers working on its development team. In contrast, OSS offers open access to its code and can benefit from contributions from a vast pool of developers outside of any company or organisation. The community-oriented nature of OSS encourages contributions, pull requests, and more. The innate capabilities of Git, such as branching, forking, and PRs, simplify asynchronous collaboration and the addition of new features. When developers introduce a feature to their preferred OSS tool, they have the option to contribute back to the original project through a pull request. This reciprocity benefits both users and developers.² By using OSS, developers can leverage well-established code, benefiting from collective wisdom. They can study, understand, and adopt the best practices they uncover, thereby gaining valuable insights at no cost.

In essence, OSS facilitates the crowdsourcing of development and troubleshooting efforts.³ Moreover, engaging community members to craft detailed documentation and set clear contribution guidelines could aid in the internal onboarding of new team members, potentially enhancing employee retention and expediting onboarding processes. For developers, OSS participation can be a valuable part of their resume as proof of their capabilities in collaboration and development, which many developers recognize.⁴ Many companies even favour development experience in an OSS context over equal experience in a proprietary one.⁵ Likewise, presenting oneself as an OSS community member can be an avenue for recruiting developer talent.^{6,7}

¹ Robles; González-Barahona (2012), P. 7–11

² Sharma (2022), P. 114–115

³ Sharma (2022), P. 112–113

⁴ Wu; Gerlach; Young (2007), P. 259

⁵ Taft (2021b)

⁶ Preston-Werner (2011)

⁷ Senz; Nagle (2018)

Finally, allowing employees to participate in OSS development has been shown to increase their productivity and facilitate learning.¹

2.4.4 Lower Costs

Developer time is valuable to a company, both through its scarcity and the relatively high salaries of software engineers; leveraging OSS can prevent unnecessary duplication of effort, thus conserving this resource.² While many OSS offerings are entirely free, even commercial OSS products can be used for free when the paid offering is not needed. When the OSS offering follows an infrastructure-based BM, self-deployment becomes possible as well.³ Utilizing contemporary technologies such as containerization and high-reliability clusters, self-deployments can match the reliability, security, and uptime of managed services, often at a reduced price, particularly when considering 'per-seat' charges or subscriptions.⁴

2.4.5 Transparency and Trust

From a user perspective, the heightened transparency in OSS is clear – the code is out in the open. They can trace every modification, understanding who made changes, their reasons, and the underlying mechanics. Conversely, there are many potential reasons to be sceptical of proprietary software, namely security⁵ and privacy controversies⁶, unexpected price inflations⁷, questionable company practices⁸, and instances of product neglect⁹. Users place their faith in OSS due to rigorous peer reviews and a highly motivated community.¹⁰

¹ Nagle (2017), P. 26–31

² Balter (2015)

³ The different types of OSBM are explained in detail in Chapter 4.

⁴ The Kubernetes Authors (2023)

⁵ Free Software Foundation (URL)

⁶ UpGuard (2023)

⁷ Clark (URL)

⁸ Lim (2021)

⁹ Ogden (2024)

¹⁰ Morgan; Finnegan (2014), P. 230

Linus's Law further supports this, suggesting that well-scrutinized and tested code is innately more reliable than strategies based on security through obscurity.¹

For developers, adopting an open-source approach can significantly boost trust, especially for smaller firms that face potential business risks. Such companies, by relying solely on proprietary software, could leave users stranded if they fold. Open sourcing acts as a sign of goodwill and fosters open communication, crucial in the software realm, particularly with emerging or nascent technologies.²

2.5 Limitations of Open-Source Software

The same two-sided approach is followed in examining the limitations and potential pitfalls of OSS.

2.5.1 Lack of Support and Danger of Abandonment

A significant number of open-source projects operate under the stewardship of a single maintainer or a very small group of contributors: A low estimate for the percentage of OSS projects maintained by a single individual is 23 percent³, while 94 percent have less than ten maintainers⁴. This heavy reliance on a very small group of developers poses a considerable risk; if these maintainers decide to step away or are unable to continue their work, the project faces potential stagnation or even abandonment, which can have devastating consequences on other projects that rely on them.

For users, this can mean a sudden lack of updates, security patches, or answers to critical queries. Depending on an open-source software that's at risk of being abandoned or lacks regular maintenance can have serious implications, especially if it's integral to their operations or infrastructure.

¹ Jones (2006)

² Senz; Nagle (2018)

³ Bressers (2022)

⁴ Synopsys (2022), P. 19

From a developer or company's standpoint, the abandonment of an open-source project could tarnish the organization's reputation and erode trust within the community. Winding down an open-source project isn't as simple as just ceasing updates; there are community expectations to manage, potential transition plans to communicate, and considerations for those who have built dependencies on the project.^{1,2} Moreover, while having a strong user base provides invaluable feedback, new ideas, and even contributions, it comes with potential risk: the possibility that the community could abandon the project. This loss means a sudden drop in feedback, testing, and community-driven enhancements. This could be a significant setback, especially if the project was started to cultivate an ecosystem around a product or service.

2.5.2 Quality and Feature Inconsistency

For users of OSS, the quality of the product they receive can vary across time and feature dimensions. Especially smaller OSS projects lack the strategic guidance a company might apply to their proprietary products. Important contributors leaving the project, and new inexperienced ones taking their place, can worsen this over time. In the absence of a legally binding list of software requirements, there is no leverage on the developers either.

From a developer's viewpoint, managing an open-source project can be akin to orchestrating a decentralized team, where contributors come with varying levels of expertise and visions for the product. While many contributions can enhance the software, ensuring consistent quality and feature alignment is a challenging task. Every contribution requires a thorough review to ensure it aligns with the project's standards and goals.

¹ phendrenad2 (2021)

² The Linux Foundation (URL)

2.5.3 Loss of Control

Adopting an open-source solution as a cornerstone within an organization carries inherent risks. A heavy reliance on a specific open-source component implies that changes to this component, whether they be functional alterations or discontinuations, can ripple through an organization's systems. For instance, when an open-source project opts not to support a particular feature or protocol, users find themselves at a crossroads: either adapt to the change, divert resources to create a workaround, or face potential compatibility issues.¹ Unlike proprietary solutions, which may offer more predictable developmental trajectories, open-source could sometimes leave users in reactive stances, especially when their internal applications heavily depend on such components.

From the developer's viewpoint, open-sourcing a product introduces its own set of challenges. In the proprietary domain, a product's distinct features and capabilities can be closely guarded as trade secrets; conversely, the very nature of open-source means that competitors can fork, resell, or offer services around the original software without providing any direct benefit to the initial developer. Additionally, having the product's code exposed to countless scrutinizing eyes demands stringent security protocols and a structured approach to vulnerability disclosure. While this open model can enhance product security through collective scrutiny, it requires companies to be constantly vigilant and proactive.

Finally, when software is introduced to the OSS community, the impact on this community as a stakeholder in the development process must be considered. While community engagement can spur innovation and enhance product quality, it could also result in alterations to the software's vision or development trajectory. Hence, community management and setting clear expectations and guidelines become pivotal.²

¹ Mayer (2023)

² The Linux Foundation; The Harvard Laboratory for Innovation Science (2020), P. 35

3 Open-Source Licenses

As outlined in Chapter 2, licensing is at the core of the open-source paradigm. To further strengthen our understanding of OSS dynamics, before delving into the research, we quickly examine the role of different license types in OSS, list the most widely used licenses and differentiate between them.

3.1 License Types

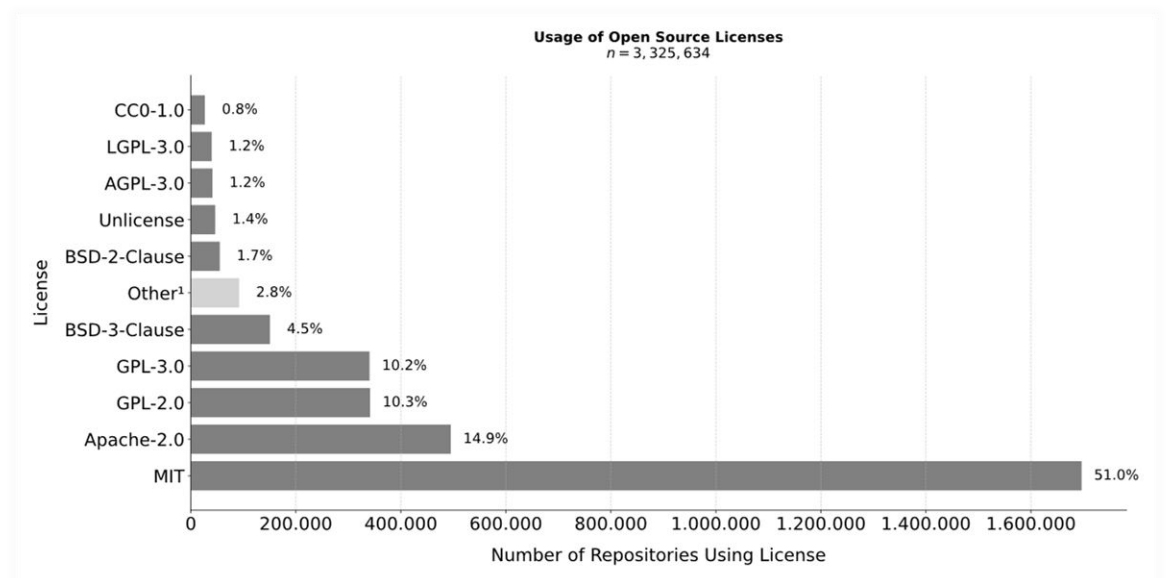


Figure 1: Usage of Open-Source Licenses in 2020

The data in Figure 1 was extracted from the *libraries.io* Google BigQuery public dataset and visualized with *matplotlib*.^{2,3}

When analysing software licenses, it is useful to classify them according to three parameters:

- Permissions: Rights granted by the license,
- Obligations: Acts mandated by the license to obtain these rights, and
- Prohibitions: Acts that the license forbids.

All three of these are highly important for open source to function.⁴

¹ *Other* comprises uncommon and non-standard licenses.

² Nesbitt; Nickolls (2018)

³ Code and assets are accessible at https://github.com/lukasgabriel/open_for_business

⁴ Riehle (2019), P. 60

The landscape of open-source licenses spans a broad spectrum, ranging from highly permissive to very restrictive. The choice of license determines how the software can be used, modified, and distributed. The nature and intentions of a project often dictate this choice.

3.1.1 The Concept of Copyleft

Firstly, the term ‘copyleft’ needs to be examined. It describes an obligation imposed by some licenses that requires any derivative work of the original to be released under the same terms as the initial license.¹

The FSF explains the term ‘copyleft’ as follows: *“Proprietary software developers use copyright to take away the users' freedom; we use copyright to guarantee their freedom. That's why we reverse the name, changing ‘copyright’ into ‘copyleft.’”* The practice of ‘copylefting’ code makes the software and the freedoms that come with it “legally inseparable.”²

In essence, this obligation mandates every “downstream” project to be open source as well. This has caused some organizations to worry about losing control of their intellectual property rights when incorporating copyleft code into their products.³

3.1.2 Permissive Licenses

Permissive licenses are often considered the ‘freest’ in terms of the liberties they provide to end-users. They tend to have minimal conditions, meaning they don't place substantial restrictions on how the software is used, modified, or distributed.⁴

¹ Free Software Foundation (2022)

² Free Software Foundation (2022)

³ Riehle (2019), P. 60

⁴ FOSSA (2021a)

Some of the key characteristics of permissive licenses are:

- Freedom to use: End-users can use the software for any purpose, be it personal, commercial, or educational.
- Minimal obligations: There are typically no strong mandates on re-distributions, such as retaining the original license.
- Inclusion in proprietary projects: A significant feature is that software under permissive licenses can be part of proprietary offerings without requiring the whole project to become open source.

Prominent examples of permissive licenses include the MIT License, the BSD license, and the Apache License.¹

3.1.3 Restrictive Licenses

On the other end of the spectrum, copyleft (as described in 3.2.2) licenses – often termed restrictive or protective – have specific conditions that ensure derivative works remain open-source.² Their main attributes include:

- Obligation to maintain freedom: Any derivative work or modification must be distributed under the same license as the original software.
- Preserving open-source integrity: By ensuring modifications remain open, these licenses keep the software's open-source nature intact through generations of development.

The GPL series of licenses are the most notable representatives of this category.

3.1.4 Public Domain Licenses

Beyond the realm of typical open-source licenses lies the concept of 'public domain' or "No Rights Reserved." Instead of merely simplifying license terms, public domain licenses such as 'Creative Commons Zero' (CC0) aims to eliminate them entirely.

¹ FOSSA (2021a)

² Maltceva (2023)

By using such licenses, creators effectively release their work into the public domain, relinquishing all their copyright and related rights.¹ This maximizes the freedom to use, modify, and distribute the work.

However, it's important to differentiate between truly placing a work in the public domain and using a license that merely mimics this effect. While CC0 comes closest to the public domain gesture, it doesn't equate to a global public domain status due to differing copyright laws across countries. For this reason, many such licenses include 'fallback clauses' to ensure that there is a minimum level of openness the license can fall back on.² Furthermore, a clause disclaiming any warrant ('as-is' clause) is usually included for the legal protection of the project authors and contributors. Popular 'public domain' licenses include 'The Unlicense' and CC0.

3.2 Open-Source vs. Source-Available Licensing

Chapter 2.1 describes the criteria which software needs to meet in order to be called open source. When source code is released under a license which does not grant the freedoms required by the OSD or the FSF, it is often called source-available software.³ Notable *source-available* licenses include the *Server Side Public License* (SSPL)⁴, the *Commons Clause*⁵, and the *Business Source License* (BSL)⁶; these licenses have been the topic of lively discussion since their introduction in 2018 and 2016 respectively.⁷ Both the BSL and SSPL were introduced to limit the conditions under which a rival company could profit from the OSS that is released under it.^{8,9}

¹ Creative Commons (URL)

² Brock (2022), P. 17–18

³ United States Department of Defense (2021)

⁴ MongoDB Inc. (2018)

⁵ FOSSA (URL)

⁶ MariaDB (2018)

⁷ Krazit (2018)

⁸ MariaDB (2018)

⁹ Dadgar (2023)

The BSL accomplishes this by prohibiting “*offering the Licensed Work to third parties on a hosted or embedded basis which is competitive with [the licensor’s] products*” for four years after its release.¹ The SSPL takes a different approach – sometimes called ‘extreme copyleft.’ It mandates anyone who offers the licensed work to a third party as a service to release the entirety of their source code which is required to run this service under the SSPL.² If a company uses a SSPL-licensed product in their service offering, it must release its full suite of server-side software along with it under SSPL. This clause aims to discourage service providers by forcing them to publicize their trade secrets, making their services reproducible by competitors.³

This alternative to OSS licenses emerged as a response to the rapid commercialization and monetization of open-source software by cloud service providers without adequate compensation or recognition to the original developers.⁴ The rise of cloud infrastructure and platform providers, which could use open-source software to offer lucrative managed services without contributing back to the community or the original software’s authors, resulted in challenges for financial viability of the BM behind the software. These source-available licenses thus aimed to provide a middle ground, balancing the principles of software freedom with protective measures against potential exploitation by dominant industry players.⁵ Since the restrictions imposed by source-available licenses infringe on Section 6 (‘No Discrimination Against Fields of Endeavor’) of the OSD, they are no longer considered open-source licenses.⁶

¹ HashiCorp (2023)

² Smith (2019)

³ MongoDB Inc. (2018), sec. 13

⁴ Lardinois (2018)

⁵ Banon (2021)

⁶ Vaughan-Nichols (2019)

4 Open Source Business Models

For someone unfamiliar with the intricacies of OSS, the prospect of developing a for-profit business model based entirely on an open-source offering might seem counterintuitive.

One might ask the – entirely valid – question “*If anyone can just download your product for free, use it as they see fit, and even modify and redistribute it, why would they pay you any money?*” In reality, there are many different answers to this question, which manifest in different OSBMs. We conduct an overview of these models and their implementation in the business landscape, examine monetization strategies, and highlight their benefits and limitations, and consider pertinent examples.

4.1 Community vs. Commercial Open Source

OSS can be classified in numerous ways across different dimensions. The first distinction to be made is between *commercial* and *community* OSS:

Community OSS reflects the traditional form of OSS in which a community of volunteers develop software together; control of this software is distributed between a group of stakeholders – individuals, companies, and stewardship organizations – from this community. While companies can play an important role in the governance of this community, and benefit from the software in several ways, they have no claim to ownership over it.

Commercial OSS is software which is released under an open source license, but ultimately owned and controlled by a company with the purpose of profiting from it. It is also referred to as *single-vendor commercial OSS*.^{1,2}

Whether a project is *community* or *commercial* OSS is a strong indicator on the efficacy of different business models taking advantage of it. To stay true to our research objectives, we only consider Commercial OSS projects in our analysis.

¹ Capra; Wasserman (2008), P. 1–2

² Riehle (2012), P. 1–5

4.2 Taxonomy of Open-Source Business Models

To further classify OSBMs, we draw upon the business model taxonomy as theorized by Al-Debei et al., particularly the V^4 business model dimensions.¹ Additionally, we utilize a simplified version of the *Taxonomy of Open-Source Business Models* developed by Duparc et al., which adapts the V^4 model to the OSS context.² This classification is used in Chapter 6 as a framework to compare the products and organizations of our case studies and analyse how different characteristics of OSBM manifest in them.

		Dimension	Characteristic							
v1	Proposition	D1	Free Offering	Software	Platform	Infrastructure	Service	Physical Product	None	
		D2	Commercial Offering	Software	Platform	Infrastructure	Service	Physical Product	None	
		D3	Customer Value	Functional	Relationship	Co-Creation	Brand	Social		
		D4	Customer Segment	Business	Public Sector	Academic Sector	Consumer			
v3	Network	D5	Community Approach	Symbiotic	Commensalistic	Parasitic				
		D6	Communication Channel	Docs	Events	Online Forums	Social Media	Social Coding	Mailing Lists	
		D7	Governance Structure	Centralized	Balanced	Decentralized				
v2	Architecture	D8	Source Code Provider	Companies	Stewardship Orgs	Individuals				
		D9	Licensing	Proprietary	Source-available	Permissive	Copyleft			
		D10	OSS Component	Complete	Core	Complementary	None			
v4	Finance	D11	Financial Purpose	Commercial	Quasi-commercial	Non-commercial				
		D12	Revenue Mechanism	Direct-sale	Indirect-sale	Funding				
		D13	Revenue Model	Subscription	Freemium	Fees	Pay-per-use	Advertisement	Passive Incomes	Donations
		D14	Price Base	Users	Time	Transactions	Downloads	Resources	Sponsors	

Figure 2: Representation of the OSBM Taxonomy in a Morphological Box

¹ Al-Debei; Avison (2010)

² Duparc; Möller; Jussen; et al. (2022)

In certain dimensions, we have adapted the taxonomy from the original work to better suit the context of our research. For example, we have omitted the original ‘*Dimension 12: Platform type*’ and ‘*Dimension 13: Boundary Resource*’ as those offered little to no distinctions between our observed cases. The ordering of the V⁴ dimensions in the original taxonomy deviated from the original ordering which Al-Debei et al. provided; we use the adapted ordering in our application, in which ‘*Value Network (V3)*’ is the second dimension and ‘*Value Architecture (V2)*’ is the third.

Value Proposition

The Value Proposition component of a business model is described by Chesbrough and Rosenbloom as “[...] *the value created for users by the offering based on the technology [...]*”¹ Duparc et al. adapt this definition to “[...] *the organization’s ability to conceptualize a product offering that delivers value to a relevant customer segment through it.*”²

Since OSS always contains a component (the source code) that is offered free of cost, this **Free Offering** needs to be complemented by a **Commercial Offering** to form the value proposition.³ The nature of these offerings lies at the core of the business model and can take various forms:

- **Software:** The software itself.
- **Platform:** Creation of a platform to grow the userbase and facilitate network effects which benefit other products.
- **Infrastructure:** Offering of infrastructure services which are complemented by the OSS.
- **Service:** The offering of value-added services such as support, consulting, and training for the OSS.
- **Physical products:** The sale of physical products which synergize with the OSS.

¹ Chesbrough (2002), P. 533

² Duparc; Möller; Jussen; et al. (2022), P. 732

³ Duparc; Möller; Jussen; et al. (2022), P. 732–734

These offerings are non-exclusive and appear in various combinations in the real world. Apart from the physical products, each of these offerings can be free, paid, or both.

The next dimension is that of **Customer Value**; it can be divided into *functional, relationship, co-creation, brand, and social value*.

The final dimension of the value proposition are the **Customer Segments** targeted by the business model; for OSS, these segments are *Business, Public Sector, Academic Sector, and Consumer*.

The permutation of these components embodies the logic behind the business model's value proposition.

Value Network

The Value Network of the business model outlines how the stakeholders interact with each other and how decisions about the product are made.¹ Since OSS, by its definition, relies on community and openness to function, the value network plays a critical role in the success of OSBMs.²

The first dimension of the value network as outlined by Duparc et al. is the **Community Approach**; they build upon the work of Dahlander and Magnusson, which divide firms into *symbiotic, commensalistic, and parasitic* towards the open-source community.³ Firms with a symbiotic approach collaborate with the community, ensuring mutual benefit. Commensalistic firms utilize the resources of the community but don't necessarily contribute back, while parasitic firms might exploit the community for their gain, potentially harming it in the long run.

¹ Duparc; Möller; Jussen; et al. (2022), P. 734

² Chesbrough; Appleyard (2007), P. 60–61

³ Dahlander; Magnusson (2005), P. 487

The second dimension is the usage of **Communication Channels**. Commonly observed channels of OSBMs are *Documentation, Social Events, Online Forums, Social media platforms, Social coding platforms, and Mailing Lists*.

The final dimension is the project's **Governance Structure**, which can either be *centralized, balanced, or decentralized*; it determines how decisions are made in the project, whether by a single entity, a balanced group, or through a broad-based community consensus approach.

Value Architecture

The Value Architecture of an OSBM refers to the technological and organizational infrastructure on top of which the OSBM is operating.

Its first dimension, the **Source Code Provider**, describes who is responsible for the creation of the source code itself. This can be *individuals, non-profit steward organizations, or for-profit companies*; this dimension mirrors the definitions 'community' and 'commercial' OSS as described in Chapter 4.1.

The second dimension is the project's **License**. This dimension is explored in detail in Chapter 3.

The final dimension is the **Open-Source Component** – the part of the product which is open-source. It can range from the *complete product, to the core components, complementary components, or none*.

Value Finance

The Value Finance defines all attributes of the OSBM which relate to the generation of income and the factors which influence the pricing.

The foundational dimension of the Value finance is the project's **Financial Purpose**; this can either be *commercial*, *quasi-commercial*, or *non-commercial*. Quasi-commercial can refer to two types of special cases: Firstly, products which generate only insignificant amounts of revenue but have the potential or intention to be commercially viable in the future can be called quasi-commercial. Alternatively, products which are non-commercial but are developed with a professional background can also be referred to as quasi-commercial. Such products might be developed with exogenous motivations such as developing or controlling a specific market segment or eroding a competitors' profit potential in that product segment.

The **Revenue Mechanism** refers to the logic behind the revenue generation. This can either be to generate revenues through the *Direct Sale* of the product or of services directly coupled with it, through *Indirect Sales* of ancillary offerings, or through *Funding*.

The **Revenue Model** refers to the specific method a company uses to implement its revenue mechanism, determine how it will charge its customers and earn income. The considered revenue models are *subscriptions*, *freemium*, *fees*, *pay-per-use*, *advertisement*, *passive incomes*, and *donations*.

Finally, the **Price Base** is the measured quantity that is the basis for the actual price charged. These quantities can be *user entities*, *time*, *transactions*, *downloads*, *resources*, and *sponsors*.

4.3 Archetypes of Open-Source Business Models

In the dynamic landscape of open-source software, a multitude of business models have emerged to navigate its unique challenges and opportunities. This Section delves into select archetypes of OSBMs based on the taxonomy outlined by Duparc et al. Our aim is to highlight both successful and unsuccessful instances of these archetypes, thereby providing insights into their respective strengths and weaknesses.

4.3.1 Infrastructure Business Model

The *Infrastructure BM* operates by delivering software solutions aimed at driving sales through related hosting services.¹ This approach can be intricate for several reasons; primarily, the hosting services integral to the model can be easily mimicked by competitors, potentially those with more attractive pricing or superior economies of scale.² Consider the dominance of hyperscalers such as Amazon Web Services (AWS) and Google Cloud Platform (GCP) in contrast to other, smaller players in the market.

Furthermore, when an OSS product is readily self-hostable and easy to deploy, both users and administrators are incentivized to opt for self-hosting. This choice not only offers cost savings but also promises enhanced control, flexibility and the option of private data storage.

A concrete example of a company navigating the intricacies of this model is Bitwarden, an open-source password management service.³ While individuals and organizations can host Bitwarden on their own servers, the company also offers its cloud-hosted SaaS version.⁴ This is especially beneficial for average consumers who might lack the expertise or desire to host such a critical service on their own. Given the nature of a password manager – where security and constant availability are paramount – many users, even those with the necessary technical skills, still have an incentive opt for Bitwarden's hosted solution.⁵ Furthermore, Bitwarden's cloud service is priced very competitively – including an unlimited free personal plan, and a paid plan with additional features.⁶ One of Bitwarden's motivations for maintaining an open codebase is the community's trust: many users believe a password manager's reliability hinges on its capacity for public auditability.^{7,8}

¹ Duparc; Möller; Jussen; et al. (2022), P. 739

² Stiefel, Michael; Mekeer; Dix (URL)

³ Bitwarden Inc. (2023a)

⁴ Bitwarden Inc. (2023d)

⁵ Bitwarden Inc. (2023c)

⁶ Bitwarden Inc. (2023b)

⁷ Young (2018)

⁸ Spearrin; Augendre; Taikon; et al. (2018)

The widely used *NoSQL* database *MongoDB* serves as another illustration; its parent company offers both an open-source version of its database and a managed database service known as "*Atlas*."¹ Yet, when cloud behemoths like AWS started offering managed MongoDB services, competition intensified. MongoDB Inc. responded strategically by introducing the SSPL described in Chapter 3.2.² This new license was designed to ensure companies like AWS couldn't simply offer MongoDB as a service without contributing back. It was a move to protect their revenue stream from Atlas, which prompted cloud providers like AWS to launch their own alternatives, such as Amazon *DocumentDB*, rather than using MongoDB directly.³

4.3.2 Open-Core Platform Business Model

The *Open-Core Platform BM* is fundamentally a tailored adaptation of the conventional freemium monetization strategy often seen in the software industry. At its core, this approach allows users to access a foundational, often fully functional version of the software for free – the "open core". Surrounding this core is a layer of advanced, premium features or services which are monetized. The extent of this monetized layer, in terms of the richness of features and its separation from the open core, can vary considerably across different implementations.

A prime exemplar of a successful open-core model is GitLab. GitLab offers a comprehensive open-source tool for the entire software development and operations lifecycle.^{4,5} While its core features related to source code management are freely available, GitLab monetizes advanced features such as enhanced security, performance metrics, and premium support services.⁶

¹ MongoDB Inc. (2023)

² Lardinois (2018)

³ Vaughan-Nichols (2019)

⁴ GitLab Inc. (2023d)

⁵ GitLab Inc. (2023b)

⁶ GitLab Inc. (2023c)

This strategy has allowed GitLab to foster a large, active community contributing to its open-source core while generating revenue from enterprises in need of advanced capabilities.¹

However, navigating the balance between open and proprietary components in the open-core model can be fraught with challenges. Sourcegraph, a code search and intelligence tool, once followed an open-core model but faced difficulties. Despite having a robust open-source offering, the differentiation between its free and paid versions wasn't compelling enough for a broad base of users to invest in the premium version.² This resulted in Sourcegraph reevaluating its approach and later pivoting to a more closed licensing model.³ The example illustrates the necessity of striking the right balance when segmenting features between the open core and the premium layer.

4.3.3 Professional Services Business Model

Under the expertise-centric *Professional Services BM*, the primary value proposition is the profound knowledge and expertise around an open-source product rather than direct monetization of the software itself. Exemplified by companies like Red Hat and Canonical, this model embraces the ethos of providing software that's completely open and free of direct commercial constraints. Revenue streams emerge from the in-depth expertise these companies possess about their respective products.⁴

This approach capitalizes on the unique position of the software creators who, having an unparalleled understanding of their product, are best equipped to provide advanced support, consulting, and custom solutions tailored to a product's deployment. It especially resonates with enterprise clients who prioritize reliability, security, and efficiency.

¹ GitLab Inc. (2023a)

² Slack (2023b)

³ Slack (2023a)

⁴ Okoli; Nguyen (2015), P. 6

These organizations often lack the in-house expertise to handle every nuance of complex open-source software. Hence, they lean towards external expertise for specialized support, workshops and training, ensuring compliance with various regulations, acquiring SLAs, guaranteeing long-term updates, and conducting certification exams for practitioners.^{1,2}

Linux server distributions like Red Hat *RHEL* and Canonical's *Ubuntu Pro* exemplify this model's potential.^{3,4} Given their extensive enterprise usage and their role as mission-critical assets, organizations frequently seek specialized support to ensure optimal operation and gain know-how that might not be present in-house. Enterprises value guaranteed support timelines, continuous security patches, and other auxiliary services like documentation, which assures them that their operations adhere to legal standards such as the European General Data Protection Regulation (GDPR).⁵ However, this model's success is contingent on the perceived value of the provided services and the indispensable nature of the software in question. A case in point is the downfall of some smaller Linux distributions such as Xandros and Mandriva which, despite having a solid open-source offering, struggled to find a niche in a saturated market dominated by giants like Red Hat and Canonical.

4.3.4 Open-Source Platform Business Model

The concept of a 'loss leader' is prevalent across industries. It represents the strategy of offering a product or service at a loss or negligible profit, leveraging it to boost demand for another more profitable offering. Classical instances include selling printers at low costs to subsequently profit from ink cartridges, or restaurants featuring a low-margin "kid's menu" to promote sales from the main menu.^{6,7}

¹ Krishnamurthy (2005), P. 6–7

² Red Hat Inc. (2022), P. 4–7, 12–16

³ Red Hat Inc. (2023a)

⁴ Canonical Ltd. (2023)

⁵ Red Hat Inc. (2022), P. 4–7, 12–16

⁶ Cobe (2014)

⁷ CBC Radio (2013)

In the realm of OSS, this strategy manifests uniquely. Large tech companies like Google and Meta employ extensive resources – from developers and community managers to dedicated support staff – for open-source initiatives.^{1,2} Although these projects may not be direct revenue generators, they serve as magnets, drawing customers to other profit-centric offerings, especially cloud services.³ For instance, Google's Kubernetes, a pioneering open-source container orchestration solution, while being entirely free, is closely affiliated with its managed counterpart, Google Kubernetes Engine.⁴ Duparc et al. coined this strategy the '*Open-source platform BM*', encapsulating it as the deployment of an open-source platform to magnetize users and catalyse network effects.⁵

In a more nuanced variation, some firms delve into OSS not primarily for direct sales or aligned services but to strategically address a market void. The return on investment isn't necessarily immediate profit, but rather long-term influence, market direction, publicity, or a blend of these objectives. Another motivation might be the commoditization of a specific product or service such that competitors can no longer profit from them. Koenig refers to these as "The Patronage Strategy."⁶ Microsoft's Visual Studio Code exemplifies this approach: Visual Studio Code is a free, open-source code editor that offers features such as debugging, syntax highlighting, intelligent code completion, snippets, and code refactoring.⁷ It was developed by Microsoft as a more approachable and streamlined alternative to its full-featured *Visual Studio* Integrated Development Environment (IDE), with the ability to run inside a web browser.⁸ With Visual Studio Code, Microsoft did not just present a competent IDE to the developer community; it subtly steered users towards its broader ecosystem.

¹ Open Source Contributor Index (URL)

² Google Inc. (2020)

³ Charvat; Cerba; Ježek; et al. (2009), P. 3

⁴ McLuckie (URL)

⁵ Duparc; Möller; Jussen; et al. (2022), P. 738–739

⁶ Koenig (2009), P. 4

⁷ Microsoft Inc. (URL)

⁸ Pasero (2021)

While Visual Studio Code seamlessly interfaces with other Microsoft services like GitHub and Azure, and prominently showcases its own extensions in the included extension marketplace, it doesn't impose Microsoft account creation or usage of specific Microsoft services.¹ It can be considered a strategic manoeuvre to gain market share and subtly influence the developer landscape.² A similar analogy can be drawn with Google Chrome, rooted in the open-source Chromium but predominantly integrated with proprietary Google features.³ In the terms of the OSBM taxonomy, these are good examples of quasi-commercial BMs.

4.3.5 Dual Licensing

Dual Licensing is an OSBM wherein a product is offered under two distinct licenses: an open-source license and a proprietary one. The open-source version is typically free for users, ensuring community adoption and contribution, while the proprietary version provides additional features, support, or legal assurances (e.g., waiver of the copyleft obligations of the community license).⁴ This model assumes a significant demand from larger enterprises or clients who are willing to pay for these added benefits or to avoid the restrictions of the open-source license.

A prominent successful example of dual licensing is MySQL. Many developers and small companies use the open-source version of MySQL, but larger organizations often purchase the commercial license to receive support, more features, or to sidestep the obligations of the GPL license under which the open-source variant is distributed.⁵

On the less successful side, the Trolltech company, with its Qt toolkit, faced challenges. While Qt is popular and widely adopted, its dual licensing model

¹ Taft (2021a)

² Codeium Team (2023)

³ Das (2022)

⁴ Rajala; Nissilä; Westerlund (2006)

⁵ Rajala; Nissilä; Westerlund (2006)

led to community discontent in its early days, primarily because the proprietary license's cost was prohibitive for many, and the open-source version's GPL license was restrictive for certain applications.¹ Although Trolltech later transitioned to a more community-friendly approach, the initial friction serves as an example of an unsuccessful implementation of the Dual Licensing BM.

5 Research Methodology

In the following Chapter, we provide a clear overview of our research approach. We initiate the discussion by outlining the selection process and rationale for our case studies. This is followed by an account of our strategy for public data collection, underscoring the breadth of our sources. To conclude, we elaborate on our questionnaire design and methodology, while also drawing attention to important limitations that may influence our results.

5.1 Multi-Case Study Design

This thesis employs a qualitative research approach consisting of a multi-case study examining three relevant companies in the open-source domain. These companies were chosen in accordance with our research objectives; detailed selection criteria are laid out in the following Section.

We begin by introducing the three products central to our case studies, along with their parent companies. This encompasses an exploration of their origins, market positioning, and other relevant attributes. Then, the business model taxonomy described in Chapter 4.2 is systematically applied to these cases; with this approach, we can describe their business models in a standardized and transferrable manner, allowing a comparative study of the OSBMs within the same research frame. Based on this taxonomy, we employ a morphological analysis, which aids in identifying patterns, similarities, and differences among the cases, allowing for the discernment of common

¹ Schmidt (2022)

success factors and challenges. We represent the position of the cases inside the taxonomy by using a morphological box, which allows for a visual representation of their business models. Lastly, decision-makers at the examined companies were given a questionnaire, with questions focusing on pertinent issues across the taxonomy domains and relevance to the research objectives. This mixed-method approach allows us to synergize findings from public sources and questionnaire responses and formulate meaningful conclusions and recommendations for businesses.

5.2 Case Selection Criteria

In Chapter 1, we identified that German small and medium-sized enterprises (SMEs) have a particular need for guidance concerning the effective implementation of OSS principles. To fulfil this research goal, we aim to examine companies and their OSS stories that can serve as attainable blueprints for SMEs with limited resources (capital, human, knowledge) and minimal prior experience as participants in the OSS ecosystem.

For this reason, we have carefully formulated the following case selection criteria:

1. Select companies that have significant ties to the German or European markets, either through their origin or strong market presence.
2. Exclude companies that exerted a transformative influence on the OSS landscape, as their strategies often intertwine too closely with broader market dynamics to offer delineable insights.¹
3. Prioritize SMEs² while excluding industry giants and nascent startups, as those may not offer scalable or relevant insights for our target group.

¹ *Red Hat* and *Canonical* can be mentioned as illustrative examples for this criterion: Since they fundamentally redefined the business models and practices around OSS, any judgements about their behaviour as mere *participants* in the OSS market are unsound.

² As delineated by the European Commission; see European Commission (2003)

Our case selection criteria closely adhere to Yin's principles of case study research, focusing specifically on theoretical propositions to guide case choices.¹ We aim for "literal replication" by picking cases that likely yield similar results, thus bolstering the study's evidence. By excluding outliers like industry giants, we align with Yin's recommendation for cases that are useful for analytical generalizations rather than broad statistical claims.² The objective is to offer actionable insights for SMEs – especially those in Germany and Europe – rather than providing a generalized view of OSS adoption. This focused approach ensures our findings are both informative and applicable, following best practices in case study methodology.

5.3 Public Data Collection

In our research from public sources, we employed a rigorous and systematic approach to gather meaningful data. Primary sources included established online newspapers, specialized blogs, industry analyses, and community forums that gave insights into popular opinions and expert observations on our case study subjects. Whitepapers, official company statements, and product documentations provided technical and strategic details. We ensured a strong academic foundation by referencing pertinent scientific articles. A hands-on examination of the software was conducted to understand its functionality and user experience. Quantitative data from prominent repositories like GitHub offered a window into the products' development timelines. For specific studies, we delved into legal filings and legislative texts, especially from institutions such as the European Commission, to provide a regulatory perspective.

5.4 Questionnaire

To gain an inside perspective on the research questions, we assembled a list of 11 questions to be answered by managerial staff of the case study companies.

¹ Yin (2018), P. 69–73

² Yin (2018), P. 91–94

The methods used to select and contact the participants, as well as the topics covered in the questionnaire, are outlined in the following Section.

5.4.1 Selection and Participation

The companies were contacted via publicly accessible channels such as contact forms, press contacts, and contact e-mails. Whenever a company was contacted via public channels, it was made clear that the questions were aimed at staff with managerial responsibility, preferably foundational members of the company or project. Where possible, the relevant persons were contacted directly via social media such as *XING* and *LinkedIn* or via their work e-mail. In anticipation of scheduling and time constraints of the busy individuals, the option of an asynchronous mode of interview (answering the questions in text form) was offered in addition to a synchronous interview via an online call. We hoped that this would afford the participants the most amount of flexibility and increase the acceptance rate of the invitations. After due time had passed, three participants agreed to the asynchronous questionnaire and provided their answers; unabridged versions are provided in Appendix 4 and Appendix 5.

5.4.2 Privacy and Information Disclosure

Participants were briefed on how their contributions and responses would be utilized and handled, emphasizing the confidentiality and anonymity aspects. The information was structured as follows to ensure participants' understanding and consent:

Inclusion and Attribution of Responses: Participants were informed that their responses would be included in the thesis, either as direct or indirect quotations, if they were congruent with the research objectives. While the unabridged and unedited versions of the responses would be included in the Appendix or Supplementary Materials, participants were made aware that such inclusion would strictly adhere to confidentiality norms, ensuring that no personally identifiable information would be disclosed in publicly accessible versions of the thesis.

Identity and Anonymity: Participants had the option to provide their names, which would only be included in the confidential version of the thesis submitted to the faculty for grading, ensuring non-disclosure in any publicly accessible version. Participants could also opt to provide information about their employer or position, which would be included in the public versions but without disclosing names of persons. In cases where the position could reveal identity, participants were encouraged to use broad terms like “management” or “development/engineering.”

Contact Information: The participant had the option to provide their e-mail address in case they wished to receive a copy of the finished thesis after its publication. Any email addresses provided by participants would only be used for sharing the completed thesis and would be strictly confidential, to be deleted no later than three months after the submission.

Ensuring Participant Consent: To ensure that all participants had duly noted and understood the stipulated terms and conditions regarding privacy and usage of their responses, they were required to affirmatively check a checkbox at the bottom of the online questionnaire.

5.4.3 Questionnaire Topics

The 11 questions were constructed to gather insights into specific dimensions of the OSBM taxonomy, specifically the decision-making process behind the companies' choices regarding these dimensions.

The questions can be mapped to the taxonomy as follows:

- Q2: Communication Channel, Community Approach
- Q3: Licensing
- Q4: Community Approach, Governance Structure
- Q6: Free Offering, Commercial Offering, OS-Component
- Q7: Revenue Model, Price Base
- Q10: Licensing

Q5, Q8, Q9 and Q11 are independent questions that do not directly correlate to a taxonomy dimension but are considered in the discussion and synthesis of our research.

Q1 deals with the experience of the participant; this helps to guarantee the relevance and conformity of the response to our research objectives.

The complete list of questions, is provided in Appendix 3.

5.4.4 Questionnaire Limitations

Questionnaire Response Biases: The limitations of relying on questionnaire insights involve the potential subjectivity and biases of the participants. The insights provided are contingent on the individual experiences, perceptions, and interpretations of the participants, and hence may not be universally applicable or reflective of the broader industry trends and norms. However, the triangulation of these insights with data from diverse public sources aimed to mitigate such limitations and offer a more rounded perspective on the research questions.

Limited number of responses: Due to the nature of the research, the questionnaire questions were aimed at people with strategic decision-making responsibilities and insight into the early days of the companies, which includes the managers and directors of the companies. These individuals are often hard to reach directly and have limited time available, which ultimately resulted in the volume of responses falling short of our expectations. Despite multiple attempts to contact the companies through different channels, only two inquiries were awarded a response at all. From these, three distinct responses could be gained for the study. However, we hope to balance this lack of volume with a strong focus on publicly accessible data and the other qualitative methods employed in this study.

6 Case Studies and Analysis

This Chapter begins with an introduction of our selected cases, including the background of the companies and products and a description of their products' market positions. We then move on to the application of our OSBM taxonomy to these cases, explaining our categorizations in the process. This is followed by the presentation of our findings regarding our research objectives, and the responses to our questionnaire.

6.1 Presentation of Individual Case Studies

Nextcloud

Nextcloud is a suite of client-server applications focused on providing file hosting services, effectively serving as an open-source alternative to cloud services such as Google Workspace (Drive, Docs, Mail, etc.) or Microsoft 365. Its unified platform, known as Nextcloud Hub, offers storage and management of documents, media files, contacts, calendars, and email. It further extends its capabilities with audio/video chat functionality, collaboration tools, including a collection of LibreOffice-compatible document editors, and project management features. Unlike typical Software-as-a-Service providers like Google and Microsoft, Nextcloud allows users to host data and computation on-premise, enhancing control and privacy.¹ This unique approach, coupled with a commitment to openness and a privacy-friendly data storage model, has garnered favor in the European public sector, making Nextcloud a notable success story in the open-source software landscape.²

Nextcloud started as a *fork* of *ownCloud*, a similar OSS file hosting software product, after ownCloud's founder and lead developer *Frank Karlitschek* – along with the majority of its core developers – left ownCloud Inc.^{3,4}

¹ Nextcloud GmbH (2023f)

² Directorate-General for Communications Networks; Blind; Pättsch; et al. (2021), P. 70–73

³ Karlitschek (2016)

⁴ Vaughan-Nichols (2016)

This departure was purportedly driven by a cultural mismatch between ownCloud's open-source developers and ownCloud Inc's managers, who allegedly misunderstood and infringed upon the OSS community's values.¹ Following this split, the American ownCloud Inc. had to cease business operations, while the German ownCloud GmbH was able to carry on with the development of ownCloud.^{2,3}

Since then, partly due to external factors such as the COVID-19 pandemic and the passage of strict privacy legislation in the EU, Nextcloud has been able to establish itself as a major player in its segment; in April 2018, the German Federal Information Technology Centre (ITZBund) announced that the tender for the establishment of a German federal private cloud ('Bundescloud') had been awarded to Nextcloud.⁴ It has achieved similar successes in France, Sweden, the Netherlands, and was selected as the collaboration platform for the EU-wide data infrastructure project *Gaia-X*.^{5,6}

In 2021, Nextcloud had a self-reported installed base of 400.000 instances, making it the frontrunner among self-hostable office & groupware products.⁷ Its growth continued in the first quarter of 2023, where Nextcloud was the 17th fastest growing OSS startup worldwide in terms of GitHub 'stars'.⁸

Nextcloud follows a 'Professional Services' BM, selling their 'Enterprise Edition' to organizations requiring superior support terms, training and workshops, security hardening, and more.

¹ Karlitschek (2018)

² Bhartiya (2016)

³ ownCloud GmbH (2016)

⁴ heise online (2018)

⁵ Scheuer; Kerkmann (2019)

⁶ Open Source Business Alliance (2021)

⁷ Nextcloud GmbH (2023a)

⁸ Runa Capital (URL)

Shopware

Shopware is a modular eCommerce platform developed by the Germany-based Shopware AG.

Originally founded as Hamann-Media GmbH in 2000 by then-16-year-old Stefan Hamann, the company pivoted to eCommerce software development in 2003. Notably, Shopware released an open-source "Community Edition" in 2012, emphasizing their commitment to openness and community-driven development. The company asserts that this move significantly broadened its user base.¹

As of 2022, Shopware's commercial offerings have expanded to include three paid tiers – *Rise*, *Evolve*, and *Beyond* – each providing escalating levels of features, plugins, extended support, and traffic capacity.² Shopware also diversified its service model to offer both Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) solutions in addition to their Self-hosted licenses.³ In September of 2022, Shopware became the eCommerce platform with the most users in the top 1.000 German online stores.⁴ The company cites its OSS approach as a driving force behind their good performance.⁵ Furthermore, Sebastian Hamman – Shopware's co-CEO – stated that it was their open core business model that enabled them to achieve their placement in Gartner Research's "Magic Quadrant for Digital Commerce" in 2020.⁶

Penpot

Penpot is an open-source platform for the design and prototyping of user interfaces. With its expansive feature-set and real-time collaboration features, it presents an open-source alternative to the market leader *Figma*.^{7,8}

¹ Shopware AG (2023d)

² Shopware AG (2023a)

³ Nordhoff; Shopware (2022)

⁴ Hofacker; Lanzerath (2022)

⁵ Homölle; Schlüter (2023)

⁶ Kolf (2022)

⁷ Penpot (2023b)

⁸ Bowman; Palmer (2021)

Penpot is developed by the Spanish startup *Kaleidos*, which specializes in open-source software; the company also develops *Taiga*, an open-source project management tool.¹ While the company *Kaleidos* was first founded in 2011, the Penpot project was started recently in 2019.²

The acquisition of market-leading user experience (UX) design software Figma by Adobe in September 2022 resulted in a significant migration of users to Penpot, since many developers and designers harbor dislike for Adobe and its business practices.³ According to Penpot's CEO, the number of sign-ups to the service increased by 5.600% after the announcement of the acquisition.⁴ This record growth brought the attention of venture capital (VC) investors with it: Penpot raised 8 million USD in Series A funding in September of 2022, and a further 12 million USD in February of 2023.^{5,6}

Even though Penpot is currently gearing up to evolve into a monetized OSS product under an Open-core BM, it relies solely on funding at the moment. However, Taiga is already monetized in a hybrid model which offers Free and Paid support tiers in both cloud and self-managed deployments.

¹ Penpot (2023a)

² Ruiz-Múzquiz (2022)

³ Iyer; Miller (2022)

⁴ Lunden (2022b)

⁵ Lunden (2022c)

⁶ Lunden (2023)

6.2 Taxonomy Application

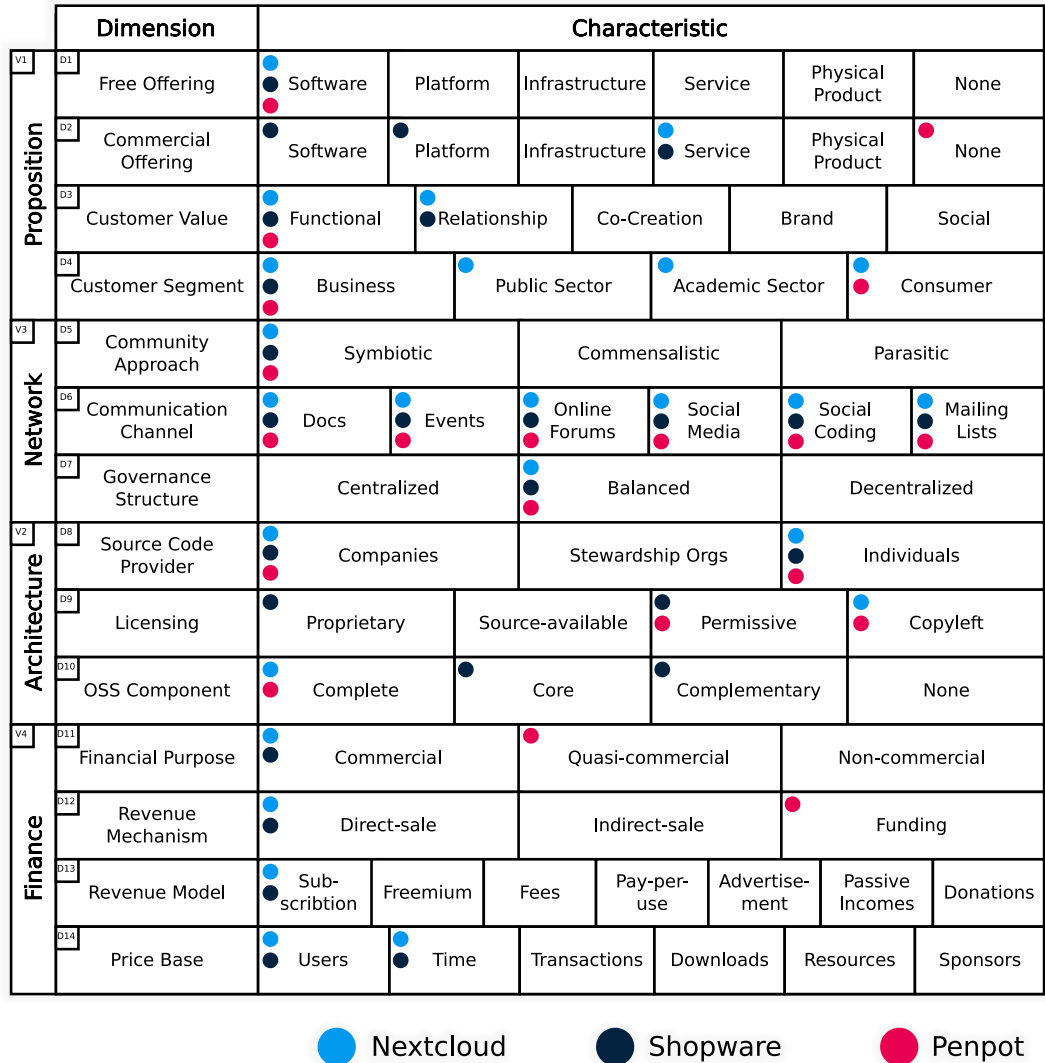


Figure 3: Placement of Companies in Morphological Box

This morphological box shows the placement of the case studies in the respective dimensions of the OSBM taxonomy. The placements represent the application of the OSBM taxonomy, which is conducted in the following Sections.

6.2.1 Value Proposition

Nextcloud

The Value Proposition of Nextcloud consists of its free and open-source Server and client components which is complemented by an ‘Enterprise Edition.’ The Enterprise Edition is based on the same open-source codebase as the free version; it differs by being pre-configured for enterprise-centric features such as legal compliance and security hardening. Additionally, it comes with enterprise support, extensive Service Level Agreements (SLAs), workshops and training, and opportunities to influence the project direction.¹ This results in Software as the *Free Offering* and Service as the *Commercial Offering*. This results in a *Customer Value* that is relationship-based. Its *Customer Segment* is diverse – as mentioned previously, it is gaining a strong footing in the public sector, as well as its already established customers in the Business sector. Although Nextcloud has many private users, its commercial offering caters almost exclusively to its business customers.

Shopware

Shopware’s offers its open-source Community Edition which includes the Shopware 6 backend, a programming interface layer, and a ‘Storefront’ Frontend. It also offers three paid plans which include enterprise features, support and optional SaaS or PaaS hosting.² In taxonomy terms, the *Free Offering* is Software, while the *Commercial Offering* is Software, Platform, and Service. *Customer Value* is generated through the platform’s functions; being an eCommerce solution, Shopware caters primarily to Business customers as its *Customer Segment*.

Penpot

For the *Free Offering*, the complete software is provided; Penpot currently has no dedicated *Commercial Offering*. However, future prospects include

¹ Nextcloud GmbH (2023d)

² Shopware AG (2023a)

the introduction of enterprise-centric features and potentially a marketplace for plugins or assets with monetization avenues.¹ Drawing parallels with Taiga, they offer a Cloud-based SaaS model that comes in both free and premium tiers; the latter includes enterprise support. The self-hosted version has an entirely open codebase, but there is a paid option for a Docker image in a managed hosting environment provided by a partner company. On top of this, there are custom enterprise deals for larger teams, which can be negotiated directly.² In the *Customer Value* dimension, both functional and relationship values are emphasized. Finally, when examining the *Customer Segment*, while the primary audience is the business sector, there is potential outreach to personal users and even those in the public sector.

6.2.2 Value Network

Nextcloud

Nextcloud's *Community Approach* can be described as symbiotic with the community since it very actively engages with it and has no proprietary-only software components. Its *Communication Channels* include a community forum, GitHub as a social coding platform, social media channels, extensive documentation, and community events.³ Nextcloud follows a balanced approach to *Governance Structure*, giving its third-party contributors some influence over the product via community communication channels and GitHub issues & pull requests, but retains the definitive decision-making capability within the company.

Shopware

Starting with the *Community Approach*, Shopware seems to stand between symbiotic and commensalistic relationships, but exhibits a more pronounced tendency towards the symbiotic nature due to its commitment to the open-source community. When it comes to *Communication Channels*,

¹ Ruiz-Múzquiz (2023b)

² Taiga (URL)

³ Nextcloud GmbH (2023c)

Shopware utilizes a myriad of tools to engage with its community. This includes conventional social media platforms, an active community forum, chatrooms facilitated through Slack, a presence on social coding platforms like GitHub, regular events to foster community interaction, and detailed documentation for developers and users alike.¹ The *Governance Structure* for Shopware is not as easily defined: While there's an openness to contributions and the company encourages community involvement, the significant proprietary component and the company's strong business-centric focus make it a challenge to determine its governance as either purely balanced or centralized. That said, it leans more towards a balanced approach, hence its placement in this dimension.

Penpot

Penpot's *Community Approach* can be characterized as symbiotic: They not only actively engage with their community but also exhibit a profound commitment to the open-source ethos, reciprocating value and fostering collaborative efforts. In terms of *Communication Channels*, Penpot embraces a comprehensive spectrum of platforms. This includes their community forum, extensive documentation, a variety of social media outlets, a regularly updated blog, participation on social coding platforms, and hosting community-centric events like the 'Penpot Fest'.^{2,3} Furthermore, they utilize video platforms like YouTube and facilitate direct communication through online chats, particularly on Gitter. As for the *Governance Structure*, it maintains a balanced approach. While there is notable community participation, especially through platforms like GitHub, and while maintainers do exert influence, the decision-making power is retained within the company.

¹ shopware AG (2023)

² Penpot (2023c)

³ Penpot (2023d)

6.2.3 Value Architecture

Nextcloud

The *Source Code Provider* of Nextcloud is Nextcloud itself, with support from the developer community from which it is accepting contributions. The server-side components of Nextcloud are under the AGPL-3.0¹ *License*; the client-side components (mobile and desktop apps) are licensed under GPL-2.0 and GPL-3.0. Since Nextcloud follows a 'Professional Services' BM, it does not retain proprietary components; hence, its *Open-Source Component* is the entire codebase.

Shopware

The *Source Code Provider* is predominantly the company itself, although the broader community actively engages in the development process. Regarding *Licensing*, Shopware employs the MIT license for the community backend and frontends, reserving proprietary licensing for their premium components. The *Open-Source Component* encompasses an open core for both the backend and frontend reference implementations, keeping paid features under a proprietary license. In addition, Shopware provides open-source offerings for ancillary resources, including Docker containers.²

Penpot

For Penpot, the *Source Code Provider* role is shared between the company and the wider community. As for *Licensing*, Penpot adopts a weak copyleft approach, emphasizing open collaboration while maintaining some controls. The *Open-Source Component* currently comprises the entire codebase.³ While there is potential for proprietary enterprise features in the future, Penpot could also adopt a model similar to Taiga, which capitalizes on hosting and support services without narrowing the open-source offering.

¹ GNU Affero General Public License: A license based on the GPL and modified specifically for software running over a network.

² Shopware AG (2023b)

³ Penpot (2023e)

6.2.4 Value Finance

Nextcloud

Since Nextcloud GmbH is a for-profit company that actively sells Nextcloud licenses, its *Financial Purpose* is commercial. Although Nextcloud briefly incorporated the sale of pre-assembled Nextcloud servers into its portfolio (through a collaboration with Western Digital Labs and Canonical), the endeavour was ultimately cancelled; therefore, Nextcloud follows a direct-sale *Revenue Mechanism* today.¹ Even though Nextcloud collaborates with select hosting partners, it does not offer hosting or SaaS itself. Its commercial component – enterprise support and services – is monetized on a time-dependent *Price Base*.²

Shopware

Shopware is a product with a commercial *Financial Purpose* that follows the *Revenue Model* of direct-sale of premium licenses. All three tiers of paid license offer SaaS, PaaS, and self-hosted modes of deployment.³ Since the terms of pricing are not made public and Shopware did not respond to inquiries, the *Revenue Model* and *Price Base* can not be determined exactly. However, classification as a hybrid of the Subscription and Pay-per-use models is likely; the Price is most likely based on User Entities and Units of Time.

Penpot

Even though Penpot has stated their intention to monetize the product in the future, there is currently no paid component. Its parent company Kaleidos has a strong ideological motivation to offer a perpetually free license of Penpot which includes most features useful to designers and developers.

¹ Poortvliet (2017)

² Nextcloud GmbH (2023e)

³ Shopware AG (2023a)

The companies' expenses are currently covered by seed and series A funding.¹ This makes the classification of the *Financial Purpose* not as straightforward; however, the category of Quasi-Commercial is most appropriate. Currently, Penpot does not have a distinct *Revenue Mechanism* or *Model*. However, drawing parallels with Taiga, there's potential for a subscription-based model in the future. As for the *Price Base*, Penpot has not established one at present. Yet, taking cues from Taiga once more, a time-based pricing structure could be a feasible approach moving forward.² However, in our classification, we position Penpot in terms of its current state.

6.3 Summary of Questionnaire Responses

In this Section, the participants in our questionnaire are introduced and their answers summarized. They are structured analogously to the mapping described in 5.4.3, which groups the questions into the dimensions of our OSBM taxonomy, with independent questions being addressed at the end.

Participants:

- Participant 1 (P1) chose to share their identity and position; they are an Employee at Kaleidos with 5 years of experience in this field.
- In the free-form entry section of the questionnaire, they noted that Q10 & Q11 were answered by the a member of upper Management at Kaleidos, who is identified as Participant 2 (P2) accordingly.
- Participant 3 (P3) chose not to share their identity; they provided their experience as a collective 8,5 years in the OSS domain, with 5 years as an engineer and 3,5 in a project management role.

In the following text, the identifiers P1, P2, and P3 are used.

¹ Ruiz-Múzquiz (2023b)

² Taiga (URL)

Value Proposition

P1 identified their business models analogously to our findings from public data; they mentioned that other monetization channels such as “marketplace fees for paid templates and libraries” could be explored in the future. The model of providing the best possible experience for free users by subsidizing development cost with paid enterprise users was mentioned as a deliberate strategy.

P1 remarked that, in their experience, offering “SaaS-only per-seat free-mium” as the Commercial Offering will not be successful at a larger scale if the OSS product is self-hostable; when self-deployment is easy, there is a clear incentive for personal users to self-host the software to save on subscription fees. In P1’s opinion, the project’s pricing and monetization scheme should be sustainable independently from the method of deployment.

P1 & P3 identified legal and security compliance features as the most promising domains for monetization of enterprise-specific features. Another area that was mentioned is that of reporting and controlling for higher-level management, integration with third-party identity providers, and centralized user and permission management.

P1 noted that, for a project to be successful with a smaller userbase, more features would need to be placed in the Commercial Offering.

P3 noted that the public sector and the academic sector were promising Customer Segments for open-source products due to political and regulatory influence favouring OSS.

Value Network

P1 indicated a preference for the use of GitHub as an internal as well as external code hosting platform; P3 shared this preference for GitHub as an external platform. The social component of GitHub “stars” was mentioned

by P1 as a valuable tool for “measuring the impact of [a] project on the developer world.” Furthermore, the discoverability of projects through GitHub’s community features was noted by P1 & P3 as valuable and impactful. GitHub’s prevalence and popularity was mentioned by P3 as an important contributing factor as well.

When asked about conflicts between financial interests and community sentiment, P1 noted that such conflicts are rare; however, P1 remarked that OSS community members are sceptical of VC investments and that transparency in decision-making is important in this regard. P3 raised the issue of source-available licensing as a current area of debate around such compromises.

P1 emphasized the importance of a Symbiotic Community Approach. P2 underlined the importance of “[understanding] the different power dynamics and conflict of interests” that arise when building a community of open-source users and contributors.

Value Architecture

All participants identified the ethical and ideological components of open-source as driving forces behind business decisions regarding licensing and open-sourcing their code. P1 explained the choice of the MPL-2.0 license with its unique attributes and their strong belief in giving back to the open-source community.

P3 noted that the topic of Licensing and License selection takes a minor role from the perspective of most users, but can be of importance for businesses that want to integrate the product with their own.

P2 and P3 gave extensive answers in regards to the topic of ‘Source-available’ licensing and licenses such as BSL and SSPL: Both stated a familiarity with this topic and mentioned the ongoing debate around it.

P2 described how the meaning of, and connotations around, 'Source-available' licensing changed since the 2000s – from being a synonym of “corporate hijacking” of the OSS community to being a response against a “game theory model [that] is unfairly broken” which originates from the OSS community itself. They attributed the root cause to the commoditisation of cloud services. In the opinion of P2, modern OSI-approved licenses are needed that can deal with the underlying issue in an OSD-compliant way. P3 noted that the “degree of risk” from this issue depends heavily on your business model and how it adds value to the customer. In their view, companies that operate a ‘Professional Services’ business model are less likely to be affected negatively than companies with a business model that is easily replicated, such as Infrastructure-based business models. P3 stated that, in the face of this issue, a change in the business model or the adoption of other revenue streams could be considered as an alternative to a licensing change. Finally, both participants remarked that they consider ‘Source-available’ licensing as a “desperate measure” (P2) and as a “last resort” (P3).

Independent Questions

Independently from our OSBM taxonomy and analysis, we asked the participants about their assessment of the perception of OSS in the software industry and among the general public. While the volume of responses limits the meaningfulness of this result, P1 and P3 both noted that the awareness of OSS increased slightly within professional IT spaces. As for the general public’s awareness of OSS, P1 and P3 gauge it as “Much more aware” and “Slightly more aware” respectively.

Finally, our last question addressed two aspects of open-source endeavours: advice for those aiming to build companies around their open-source projects and guidance for proprietary software companies considering an open-source transition.

P2 stressed the importance of an understanding of one's position in the value chain, and the potential for third-party intermediation. The stated that recognizing who "owns" the relationship with your userbase is paramount.

P3 highlighted the importance of transparency in communication with the product's customers and community members, and suggested hiring developers with experience in OSS communities as an effective way of achieving this. They also emphasized the importance of a strong focus on the product, especially in early stages. Lastly, given the current momentum of open-source, P3 noted that embracing OSS could be an avenue to attract top-tier talent, since the open-source spirit is appealing to many engineers and can set companies apart in a competitive market.

7 Findings and Discussions

7.1 Key Insights Derived from Case Studies

We present the most important insights gained from the three case studies – both from our public data collection and analysis, and from the questionnaire responses from decision-makers – divided into the four dimensions of the OSBM taxonomy.

7.1.1 Value Proposition

Free Offering: The three case studies under review exhibit a distinct approach to open-source software (OSS). Their central product is a self-hostable Free Component; infrastructure or platform components are not included in this offering. Furthermore, non-paying customers only receive community-level support that is treated with less urgency than commercial support. This model minimizes the costs incurred on the companies – since the cost of reproduction of the OSS software component is zero – while providing all of the discussed benefits of OSS participation.

Commercial Offering: Based on the questionnaire data, particularly responses from P1 and P2, a notable pattern emerges. OSS projects that focus on self-hostable software cannot rely solely on “per-seat freemium pricing”¹ as the financial backbone of their commercial offering. Instead, they must diversify their commercial offerings, integrating enterprise features, consulting, and enhanced support services. A key rationale lies in the technical demands of enterprises: administrators familiar with self-hosting may seek comprehensive control over data, processing, and user management. Offering value-added services, such as Enterprise single sign-on, reporting and auditing, and security hardening, appeals to these administrators, saving them time and ensuring system uptime. Karlitschek refers to this as “job insurance of the sysadmin”² since the final responsibility for security and uptime is placed into the hands of the external service provider. Importantly, these additions don't detract from the software's essential functions, allowing it to remain accessible to hobbyists or smaller teams. This commercial strategy, especially in an open-source environment where intellectual property is transparent, enables nuanced market segmentation. It caters to both free users – who benefit from an enterprise-subsidized product – and commercial entities that leverage a robust OSS community. These conclusions are supported by public statements made by the companies, as well as publicly accessible user feedback in forums.

Customer Value & Customer Segment: Although free users do not engage in a traditional contractual relationship with the company, they remain significant. Their non-financial contributions – feedback, community support, and advocacy – enrich the software's development and broader adoption. Functional value is the primary concern for free users, given that they don't typically receive personalized support. Furthermore, the satisfaction of these users indirectly affects the adoption rate among commercial users. For instance, the case of Nextcloud highlights the importance of considering free users as legitimate customers.

¹ See response of P1.

² Karlitschek; Nextcloud GmbH (2020), P. [25:45-26:26]

Individual users, especially those capable of self-deployment, are less inclined to pay for open-source software. This distinction becomes crucial for open-core and infrastructure-based business models.

On the contrary, professional services business models primarily target business, public, and academic sectors as their commercial clientele. These segments demand robust support, uptime reliability, and enterprise-grade security. Notably, the public and academic sectors, as evidenced by Nextcloud and the questionnaire data, are promising due to factors like funding, compliance requirements, and potential regulatory inclination toward open-source software.

7.1.2 Value Network

Community Approach: Each of the case studies adopts a symbiotic community approach, a sentiment echoed in the questionnaire responses emphasizing community interaction and transparency.

Governance Structure: The governance structures of all three case studies are balanced; none of the case studies entirely limit the agency of contributors, but keep guardrails in place and important decisions are ultimately made at the companies' discretion. However, they balance their governance by welcoming contributions and encouraging shared responsibilities. Notably – while hard to quantify – all three companies take care in following an eye-to-eye relationship with their community and contributors.

Communication Channels: All three case studies employ a plethora of communication channels. Essential resources like documentation are inherent. Besides this, they engage across a diverse range of social media and chat platforms, both mainstream (e.g., Instagram, Twitter¹, Discord) and more technical (e.g., Mastodon, Slack, Gitter, Matrix). Their outreach also

¹ To avoid confusion – especially since most of the cited sources use this name – we refer to the recently renamed Twitter by its established former name.

extends to unidirectional media like blogs. Additionally, each of the case studies conducts events welcoming both developers and users.

Emphasizing the significance of social coding platforms, all three case studies actively engage on GitHub.

Rather than merely using it as a code repository, they utilize a variety of its unique discoverability and social features. All three have published contributor guidelines and have a well-maintained landing page complemented by comprehensive READMEs. These resources undeniably simplify the orientation process for potential developers and users.

Completing this comprehensive approach to community management and patronage, two of the case studies have dedicated personnel managing community and developer relations. For instance, Kaleidos has a "Community Advocate" and Shopware employs both a "Community Manager" and a "Developer Evangelist."

7.1.3 Value Architecture

Licensing: Regarding licensing strategies, businesses might consider licenses that deter cloud providers from capitalizing on their business model, even if this veers away from a traditional open-source license. However, our research and questionnaire feedback highlight the potential merits of revisiting the business model prior to considering license alterations. Focusing on a Commercial Offering which the company can uniquely provide, i.e. in an Open-Core or Professional Services BM, can mitigate the danger imposed by intermediation.

Another important consideration in Licensing is the legal framework around community contributions. Shopware utilizes a Contributor License Agreement (CLA), which ensures contributors grant the project owner a license to use their contributions, while the contributors retain their copyright.¹ This approach can be pivotal in maintaining flexibility for future licensing

¹ Shopware AG (2023c)

changes. In contrast, Nextcloud adopts a Copyright Assignment Agreement (CAA). A CAA requires contributors to transfer their copyright to the project owner, ensuring unified copyright ownership and possibly simplifying license changes or dual licensing.¹ Notably, Nextcloud's predecessor, own-Cloud, implemented a CLA, which Karlitschek perceived as a barrier – prompting a strategic shift with Nextcloud's adoption of a CAA. Today, Nextcloud follows the 'Inbound \equiv Outbound' model of licensing, which achieves a parity between the 'inbound' license of contributions made and the project's own 'outbound' license.²

Open-Source Component: Shopware shows an example of how a product can transition to an OSBM without compromising its revenue streams. Conversely, the case of Penpot illustrates how a product can grow from a purely open-source quasi-commercial model into an open-core model, and how this can help the company shape its offering and grow its customer base.

7.1.4 Value Finance

Financial Purpose: Penpot exemplifies a strategic progression in OSS projects. It's positioned as a quasi-commercial initiative with intentions of transitioning to an open-core BM in the foreseeable future. The trajectory suggests that a robust foundation can be laid and a userbase cultivated prior to the integration of non-open or non-free features.

Revenue Mechanism & Revenue Model: Each of the three case studies operates on a direct sale model. Synthesizing this with questionnaire insights, overarching industry analytics, and current market tendencies, it becomes evident that indirect sales approaches, akin to Infrastructure BMs, may be suboptimal for emerging OSS enterprises. There's an evident challenge in segmenting the customer base and honing in on lucrative enterprise sectors. While models like open-core or professional services facilitate targeting enterprise clientele – who perceive value in niche offerings such

¹ Brock (2022), P. 117–121

² Nextcloud GmbH (2016)

as compliance features – the infrastructure BM is fraught with complications. Not only does it escalate costs (with more customers leading to augmented hosting expenditures), but it also introduces the peril of third-party mediation, as previously discussed.

Early-stage investor funding can be an impetus for OSS projects but may also pose threats to the community ethos, especially when sourced from large-scale investors pursuing ephemeral tech trends. These investors might gravitate towards "hype products" that occasionally dissipate into non-entities or pivot to contentious strategies like user data monetization. For OSS endeavors, exploring public financing and grants can be a strategic move, especially considering the challenges posed by significant cloud providers and the evolving landscape of source-available licensing.

Price Base: The determination of a project's price base is pivotal. Whether it's predicated on user entities or duration (time-based), the pricing strategy should be congruent with the business model's intent and the perceived value to the end user. The decision between user-entity and time-based pricing often hinges on the product's usage dynamics and the targeted audience's consumption patterns.

7.2 Research Question Discussion

Based on the findings in 7.1, we apply our findings towards our research questions and aim to provide answers.

RQ1: What specific management and legal strategies have proved successful for companies producing open-source software?

Due to the breadth of product and business model types in the software industry, there is no single strategy that provably leads to success. However, the following approaches toward market segmentation, product pricing, and software licensing have shown to be success factors independently of product parameters:

Diversified Commercial Offering: Optimizing the boundary between Free and Commercial offering is, in our view, the most significant determinant of OSS project success, especially in Open-Core and similar models. The viability of a Commercial Offering that is targeted solely at a very small percentage of the Customer Base depends on the scale of the project, and the degree to which these paid features can be monetized. Accurately setting a realistic and feasible boundary between Free and Paid features early on is crucial for OSS projects to succeed.

Careful User Segmentation: Recognizing and catering to both free users and commercial entities is vital. While free users contribute non-financially through feedback, community support, and advocacy, commercial users value enhanced functionalities and robust support. Tailoring offerings for distinct customer segments like individual users, business entities, and academic sectors yields better results.

Informed Licensing Choices: The choice between open-source, source-available, and libre licensing can play a crucial role. Infrastructure-based BMs, and professional services BMs might lean towards copyleft licenses to safeguard against competitive threats, ensuring that modifications remain open and benefiting the original project.

Appropriate Legal Framework around Contributions: Deciding between CLA and CAA is key. While CLAs allow contributors to retain copyright while granting the project a license to use their contribution, CAAs require contributors to transfer their copyright to the project, providing more legal control to the project owners.

RQ2: How can businesses identify and apply the most effective monetization models for their open-source projects based on their unique contexts and circumstances?

Decision-makers should begin with the OSBM Taxonomy – adapted from the OSBM taxonomy by Duparc et al. – as it allows them to decompose their business model into distinct dimensions. By using the morphological box format we have shown in our case studies, they can visualize the different placements of their BM within the taxonomy.

First, the dynamics of both Free and Commercial offerings should be analysed. By delving deep into Value Proposition and Value Finance specifically, businesses can get a clearer picture of their position in the market. It's imperative that businesses understand their Customer Segment and assess whether their Free or Commercial offerings are appealing or suitable for the respective segments. Acknowledging the distinct needs of these segments and discerning whether the provided value is more relationship-based or functional is essential.

From this foundation, businesses can determine their position in the value chain, taking into account the value they offer and what the customer ultimately utilizes. By applying this framework, businesses can strategically set the boundary between their Free and Commercial offerings. The goal should be to ensure that a considerable amount of value is presented in the Free Offering, while the Commercial Offering remains profitable and supports development operations. It's also vital for businesses to remain aware of the scaling dynamics associated with both their free and commercial offerings. They should consider the potential implications on profit when expanding their free userbase in comparison to growing their commercial userbase. Opting for a Direct-sale Revenue Mechanism is advised unless a business is confident that its ancillary Indirect-sale model isn't vulnerable to intervention by another service provider. When evaluating this, businesses shouldn't merely consider their own cost dynamics; it's important to recognize that hyperscalers operate with entirely different economies of scale.

When devising a pricing base, businesses shouldn't default to a simplistic model. They should instead deliberate on how costs might shift based on variables like an increase in users or the resources consumed by those users.

RQ3: How do open-source companies balance effective community growth and project direction while retaining control and adhering to open-source principles?

The following measures consistently appeared in our research as positive factors for sustained community growth and harmonious collaboration in OSS communities of commercial OSS projects:

Driving Community Engagement: Adopting a symbiotic community approach has been a standard among successful OSS projects. The significance of maintaining a balanced relationship with the community, characterized by an eye-to-eye relationship with contributors, is evident. Companies like Kaleidos and Shopware even employ dedicated personnel like 'Community Advocates' and 'Developer Evangelists' for effective community engagement.

Following a Symbiotic Community Approach: An effective way to nurture community growth is by adopting a symbiotic approach. This means recognizing that the community and the company are interdependent, wherein the company benefits from the community's contributions and advocacy, while the community gains from the company's resources, direction, and stewardship.

Maintaining a Balanced Governance Structure: One of the cornerstones to achieving the balance between control and open-source principles is to have a transparent and fair governance structure. While key decisions might be at the company's discretion, this should not overshadow the importance

of contributors' agency. Welcome contributions, encourage shared responsibilities, and foster an environment of trust and mutual respect.¹ Successful companies like those in your case studies ensure this by maintaining an "eye-to-eye relationship" with their community.

Engagement through Diverse Communication Channels: Facilitating open communication is pivotal. Successful open-source projects engage across a spectrum of channels, from social media platforms like Instagram and Twitter to technical forums like Mastodon and Slack. Such a diversified communication approach, supplemented by events that welcome developers and users, encourages community participation and feedback, ensuring the project remains relevant and aligned with user needs.

Active Participation in Social Coding Platforms: Platforms like GitHub are not just code repositories but avenues for community engagement. Ensuring the active use of these platforms, from addressing issues and pull requests to providing comprehensive READMEs, fosters a sense of belonging and ownership among community members. This active engagement helps in driving the project direction in a way that's both inclusive and adherent to the open-source ethos.

Regular Feedback Mechanisms: Setting up regular feedback loops, be it through community forums, periodic surveys, or open town-hall meetings, ensures that the community's voice is heard and integrated. It provides an avenue for the company to communicate its vision and for the community to offer insights, ensuring the project direction remains collaborative.

By implementing these measures, firms can hope to foster sustained and healthy growth of their OSS product communities.

¹ Carter; Groopman; The Linux Foundation (URL), P. 43–44

7.3 Potential Future Challenges of Examined Companies

In this Section, we briefly discuss potential problems that may arise for our case study companies in the future, extrapolating from current market trends.

Nextcloud

Nextcloud offers a wide array of features, but there's a potential risk of feature creep that could challenge its maintainability. Currently, there seems to be a reliance by users on the cloud features built into their devices' operating systems, e.g. iCloud for Apple devices and Microsoft 365 for Windows devices.^{1,2} This is most likely due to these manufacturers heavily promoting the use of their own ecosystems in their operating systems.³ This could place Nextcloud at a potential disadvantage, although the introduction of the Digital Services Act (DSA) in the EU might legally prohibit these practices and therefore offer a buffer against this trend.⁴

Furthermore, Nextcloud boasts an expansive plugin ecosystem, which is one of its strengths.⁵ However, if key plugins are abandoned and Nextcloud can't sustain them in-house, it might result in user disillusionment. It's also worth noting the emerging importance of AI tool integration. While Nextcloud has begun to incorporate Artificial Intelligence (AI) tools and has even launched a 'traffic light' safety classification system for plugins, continued innovation in this domain is crucial for its ongoing relevance.⁶

Lastly, Nextcloud's CEO Karlitschek stated that there have been instances in which large enterprise customers did not see the added benefits of an enterprise subscription, opting to use the community version instead.⁷

¹ Leswing (2020)

² HT Correspondent (2018)

³ Marsden; Brown (2023), P. 3–10

⁴ European Commission (2023)

⁵ Nextcloud GmbH (2023b)

⁶ Schneider (2023)

⁷ Prototype Fund n.d., P. [08:15-09:30]

If more of Nextcloud's enterprise customers decide to rely on their own expertise for support and maintenance, it could manifest in a trend that erodes Nextcloud's customer base.

Shopware

While Shopware possesses a robust feature set, its scale pales in comparison to tech giants like Shopify.¹ Since vast datasets are crucial for analytics, decision-making, and AI model training, this disparity in size could place Shopware at a disadvantage. The immense reach of platforms like Shopify not only provides them with enhanced visibility but also grants them greater access to invaluable user data. Potential remedies for Shopware could lie in public data spaces, which might level the playing field.² Shopware's recent adaptations to its pricing and subscription model is also worth noting, as not all customers will have reacted positively to these changes, as they resulted in significant price increases for some usage profiles.³ If these customers reevaluate their need for Shopware's premium features, they may migrate to the open-source Community Edition and cease to pay revenues.

Penpot

Penpot faces the challenge of retaining its user base amidst aggressive moves from competitors like Figma, whose parent company Adobe is known for pursuing lock-in strategies through its Creative Cloud ecosystem.⁴ Given the entrenched presence of these software solutions in education, coupled with their competitive student pricing, the task becomes even more daunting.⁵ Adobe, in particular, is swiftly integrating AI features which rely heavily on cloud-based compute power – resource that smaller firms like Penpot might lack.⁶

¹ BuiltWith Pty Ltd (2023)

² European Commission (2018), P. 8–12

³ webfellows UG (2022)

⁴ Weatherbed (2022)

⁵ Adobe (2023)

⁶ Weatherbed (2023)

Adobe's recent entrance into a partnership with NVIDIA further underlines its commitment towards AI features and strengthens its market position in this regard.¹ Establishing an extensive backend to support and bill users for such features is another hurdle Penpot would need to surmount.

While the allure of Penpot being free is undeniable, the absence of groundbreaking features such as AI image generation – a significant time-saver for designers – might diminish its appeal strongly. However, introducing AI-enhanced plugins could be a potential solution; moreover, Penpot is evaluating the inclusion of AI functionality already.²

Balancing their free and commercial offerings, as previously discussed and mentioned in their questionnaire response, also remains a pivotal concern for Penpot's sustained growth and relevance.

8 Conclusion

In this thesis, we provided a thorough examination of OSS history, contemporary trends, and the associated legal framework. We introduced and elaborated on the OSBM taxonomy, identifying specific archetypes and exemplifying them with cases of varying success. Through our detailed case studies, we applied this taxonomy, presented questionnaire responses, and performed a morphological analysis to highlight relationships across different OSBM dimensions. From this extensive review, we distilled key findings and formed answers to our primary research questions.

In the following Sections, we extrapolate from our research and introduce our predictions on future developments in the area of OSS. For these trends, we offer actionable recommendations for businesses, and suggest promising areas for further research.

¹ NVIDIA Corporation (2023)

² Ruiz-Múzquiz (2023a)

8.1 Future Developments and Strategy Recommendations

In this Section, we present our ideas about possible future developments in the broader space of Open Source, based on our research and findings. From these ideas, we subsequently derive strategy recommendations for businesses and organisations.

8.2.1 Increasing Openness, Decentralization, and Federation

Our case studies indicate a rising preference for alternatives to proprietary and centralized cloud-based offerings from major providers. For instance, Penpot is emerging as an alternative to Figma, Nextcloud challenges Google Workspace and Microsoft 365, Shopware is an open-source and on-premise alternative to Salesforce, and Taiga competes with Jira as an open-source project management suite. These alternatives aren't just catching up in terms of features, reliability, and security; they're often surpassing their mainstream counterparts, particularly in the areas of flexibility, user control, and privacy.

This trend isn't limited to software applications either: There is a noticeable movement in the realms of social networks with users shifting from platforms like Twitter to decentralized platforms such as Mastodon.^{1,2} Additionally, in the search engine landscape, users are turning to alternatives like Duck-DuckGo, possibly driven by annoyance with Google's increasing pollution with advertisements and its selling of user data.^{3,4} In the rapidly advancing field of AI, open-source approaches to code and datasets are present as well: Communities like Huggingface⁵ enable collaborative development of machine learning models, and AI startups like the German Jina AI⁶ develop cutting-edge AI tools under open-source terms.

¹ Lunden (2022a)

² BuiltWith Pty Ltd (URL)

³ Pike n.d.

⁴ Barry (2023)

⁵ Hugging Face, Inc. (2023)

⁶ Jina AI GmbH (URL)

Beyond software, there's an expanding horizon for open hardware as well, marked by the repairability movement and 'right to repair' legislation.^{1,2} The success of companies such as Framework^{3,4} and Fairphone⁵ represent this paradigm shift in the hardware sector.

Legislation and public funding further support the mentioned trends, especially in the EU and Germany:

- The EU's Open Source Strategy demonstrates a strong commitment to fostering innovation and collaboration through OSS.⁶
- Horizon Europe, one of the EU's key funding programs for research and innovation, has allocated substantial resources specifically to bolster European OSS initiatives.⁷
- The project Gaia-X is working towards a federated, open data infrastructure in Europe.⁸
- The European Commission has announced an extensive right-to-repair 'legal landscape' – first laws came into effect in 2022.⁹
- The German government's endorsement of Nextcloud as the 'Bundescloud' showcases its confidence in and commitment to OSS.¹⁰
- Significant public funds, such as the "Sovereign Tech Fund" and "ZenDiS", have been established to support and promote open-source projects in Germany.¹¹
- The Digital Markets Act (DMA) and DSA will significantly limit the extent in which tech corporations can exert market power and subvert competition.¹²
- The GDPR, while broader in its objectives, can be considered another milestone for the rights of individuals concerning their personal data.

¹ Tsukayama (2023)

² Ganapini (2023)

³ Framework Computer Inc. (2023)

⁴ iFixit (2023)

⁵ Fairphone B.V. (2022)

⁶ European Commission (URL)

⁷ Bolívar; Camacho; Salomon (2022)

⁸ Gaia-X European Association for Data and Cloud AISBL (URL)

⁹ European Parliament (2022)

¹⁰ heise online (2018)

¹¹ Klein (2022)

¹² European Commission (2023)

In essence, there is an overarching and multilateral move away from the centralisation of data and control under a few international tech giants.

In the foreseeable future, we anticipate a decline in the viability and prevalence of closed, proprietary, and centralized ecosystems – and a wide-ranging push towards transparency, openness, and empowerment of individuals and communities.

As evident from our research and cited previous works, developers view OSS favorably for ethical and ideological reasons; more pragmatic reasons such as enhanced job opportunities are also relevant for them.¹ Embracing OSS and leveraging developers' motivations can help tech companies weather the storm of the ever-rising shortage of IT professionals.² Beyond the topic of recruitment, rising public trust and awareness of open technologies can be attractive to potential users as well.

In summary, we recommend that businesses which do not yet participate in the OSS community via contributions or sponsorship evaluate the possibility of doing so to take advantage of the aforementioned trend. Businesses which already partake in these ways should consider deepening their involvement and, most importantly, outwardly communicate their OSS efforts to their customers, investors, and potential employees.

Effective measures to achieve this include:

- Maintaining a presence on platforms such as GitHub and publishing contribution guidelines, comprehensive documentation, and “good first issues” for first-time contributors.³
- Publicly sharing information about the companies' OSS efforts on a blog or social media.
- Attending & sponsoring OSS events, i.e. FOSDEM⁴ and OSCON⁵.

¹ The Linux Foundation; The Harvard Laboratory for Innovation Science (2020), P. 28

² Bitkom e.V (2022)

³ The Linux Foundation; The Harvard Laboratory for Innovation Science (2020), P. 28–29

⁴ FOSDEM (URL)

⁵ O'Reilly Media Inc. (URL)

- Hosting workshops, webinars, and hackathons.
- Partnerships with & sponsorships of OSS projects and foundations.
- Appointing a Community Manager, Developer Relations Manager, or OSS Outreach manager.
- Establishing an OSS Program Office.¹

Finally, companies should evaluate the possibility of receiving grants or government funding towards their eligible OSS projects.

8.1.1 Growing Importance of Social Dynamics in Open Source

The social cohesion of an OSS community often determines its success just as much as the quality of the code. Recent years have witnessed OSS communities coming together more often to voice their concerns or take a stance on issues. Instances like the debate surrounding Richard Stallman's position in the Free Software Foundation underline the maturation of community voices in OSS.^{2,3,4,5} The increased emphasis on social issues and ethics in OSS, while commendable, is a double-edged sword for businesses: While it can lead to more inclusive and ethically grounded projects, it also has the potential to cause rifts and fierce disputes over ideological issues that are hard to resolve.⁶ This can lead to community splits, forks in projects, or the alienation of certain contributors.^{7,8} Conversely, fostering a welcoming and inclusive community can expand the pool of potential users and contributors, especially towards those that are marginalized.⁹ Companies that understand the importance of community interaction, steering and management can have an advantage in the face of the growing importance of these topics.

¹ FOSSA (2021b)

² Brodtkin (2021)

³ Varghese (2021)

⁴ Claburn (2021)

⁵ De Blanc; Freitas; Garrett; et al. (2021)

⁶ Krill (2017)

⁷ Robles; González-Barahona (2012), P. 9–10

⁸ Leigh (2014)

⁹ Carter; Groopman; The Linux Foundation (URL), P. 5–7

To achieve this, we raise the following recommendations:

- Establishing clear contribution guidelines that provide a transparent roadmap for newcomers and highlight the project's objectives, the steps to contribute, and the standards to adhere to.
- Appointing a Community Manager (as mentioned in 8.2.1) who acts as a bridge between contributors and the core team and ensures that community concerns are addressed promptly and professionally.
- Implementing a robust code of conduct that sets behavioural standards and provides a framework to address and resolve conflicts.¹
- Following a zero-tolerance policy towards hate and discrimination.
- Prioritize hiring individuals with OSS experience, as they can be instrumental in following community-driven strategies.
- Offering mentorship programs for early-stage contributors.²

8.1.2 Increasing Pressure from Third-Party Service Providers

There is currently a conflict between preserving the open-source ethos and securing commercial viability against potential threats from commoditized cloud services. The topic of Source-available licensing has been mentioned and analysed at multiple points throughout this thesis. While it is not immediately relevant to our case studies, we use this Section to offer avenues for businesses to prepare for, and adapt to, the challenges associated with this industry trend.

At the time of writing, the situation around the Infrastructure-as-Code tool Terraform and its licensing change toward the source-available BSL licensing scheme is rapidly developing. As a response to HashiCorp's move towards BSL licensing for Terraform, the fork 'OpenTofu' has formed and is quickly gaining traction; it is published under the weak copyleft MPL-2.0 license – the same license that Terraform used until recently.³

¹ Carter; Groopman; The Linux Foundation (URL), P. 35–36

² Carter; Groopman; The Linux Foundation (URL), P. 39–40

³ OpenTofu (2023b)

Its manifesto has amassed 158 company backers, 781 signatories, and the repository has reached 36.000 'GitHub stars'.¹

Most importantly, the project has officially been placed under the stewardship of the Linux Foundation and aspires to become a member of the Cloud Native Computing Foundation as well.^{2,3} It is motivated by the belief that tools which are critical for modern cloud and infrastructure automation must remain fully open-source.⁴ According to OpenTofu, several companies have already committed to financially support full-time software engineers working on the project.⁵ Meanwhile, HashiCorp argues that the licensing change was a move made out of necessity due to cloud providers exhibiting parasitic behavior towards the project and violating "the spirit of open-source."⁶ It is evident how this situation is similar to the licensing changes of Elastic and MongoDB, both of which also adopted Source-available licensing after their revenue model was severely challenged by cloud providers such as AWS.^{7,8}

Some prominent voices argue that such licensing changes are a "necessary evil" to protect the commercial viability of open-source projects which rely on the provision of platforms and infrastructure; others refer to the abandonment of OSI-approved licensing as a way of "saving open-source by killing it."^{9,10} But many agree that this issue stems from the fact that the legal framework around OSS is not equipped to deal with today's market in which cloud services have become heavily commoditized.^{11,12}

¹ OpenTofu (2023c)

² Miller (2023)

³ Dierking (2023)

⁴ Dinu (2023)

⁵ OpenTofu (2023a)

⁶ Dadgar (2023)

⁷ Banon (2021)

⁸ Lardinois (2018)

⁹ See Questionnaire responses 2 & 3

¹⁰ Karlitschek; Nextcloud GmbH (2020), P. [31:20-32:02]

¹¹ Oberhauser (2021)

¹² Zaitsev (2023)

They claim that new OSI-approved licenses, that manage to protect the interests of open-source providers against cloud and managed service providers, are necessary.⁸ However, no such middle ground is currently available, and the conflict between OSS providers and hyperscalers stands to worsen.

From these developments, we derive the following recommendation: Given the rapid developments in the realm of Source-available licensing, businesses must adopt a vigilant and informed stance. Companies need to evaluate and understand their position in the value chain and consider the potential for third parties to intermediate their customer relationships. They should also proactively bolster their perceived value to their customers and aim to continuously strengthen the bonds to their users and developers. When faced with the necessity of a license change, they should consider the currently available options – i.e. SSPL, BSL, Commons Clause, and ‘Faircode’ licenses – carefully.

8.1.3 Worsening Software Supply Chain Security

Software supply chain attacks, as introduced in Section 2.3.2, underscore the pressing need for businesses to maintain a robust security posture. Ensuring consistent security within the open-source community can be challenging due to various actors, motivations, and lack of standardized practices. The ISO/IEC 5230 standard – known as OpenChain – offers a set of rules designed to achieve consistent software supply chain security. It provides a comprehensive guideline that businesses can follow to fortify their software supply chain, thereby reducing the risk of potential vulnerabilities.¹ However, a 2021 survey of German companies revealed that 51% were unfamiliar with this standard.²

¹ The OpenChain Project; The Linux Foundation (2023)

² Bitkom (2021), P. 39

In the domain of preemptive security, platforms like HackerOne have emerged to offer the facilitation of bug bounty programs between businesses and individuals who discover vulnerabilities in their software.¹ They incentivize ethical hackers to discover and responsibly report vulnerabilities by offering monetary rewards.² Such programs not only promote ethical behavior but also serve as a testament to a company's commitment to security. The legal actions taken by the German company "Modern Solution" against a hacker who responsibly reported a vulnerability underscores a critical gap in many countries' legislation, and serves as a worst-case example for this topic.^{3,4} The fear of legal repercussions can deter hackers from reporting, potentially leading to unaddressed vulnerabilities or leaks.

In light of these current topics, we propose the following recommendations:

- Adopt OpenChain Standards: Implementing and enforcing OpenChain rules can set a foundation for consistent security practices.
- Establish Transparent Security Protocols:
 - o Publish clear Security Disclosure Policies.
 - o Implement a security.txt file, detailing contact information and preferred methods for vulnerability reporting.⁵
- Offer a Legal Safeguard for Ethical Hackers by asserting that responsible vulnerability disclosure is permissible and encouraged.⁶
- Initiate Bug Bounty Programs: Utilize platforms like HackerOne to incentivize the discovery and responsible reporting of vulnerabilities.

8.2 Suggestions for Future Research

Throughout our work on this thesis, we have identified the following areas in which further research may be promising and able to provide valuable new perspectives and discoveries to academia and practice:

¹ HackerOne (URL)

² Cimpanu (2020)

³ Tremmel (2022)

⁴ Scherschel (2023)

⁵ Shafranovich; Foudil (URL)

⁶ Nidecki (2023)

Community and Social Dynamics in Coding Platforms:

We see a research gap concerning the effect of social dynamics and social coding features on the success of OSS projects.

Licensing Dynamics in Open Source:

We identify a need for research into the consequences of deviating from OSI-approved licenses: Quantifying the balance between business advantages and potential OSS user attrition due to such shifts would be very valuable for businesses in this situation. Furthermore, understanding the actual versus perceived threats of open-source forks can shed light on their implications for companies. We also reiterate the aforementioned lack of future-proof OSS licenses and suggest this topic as a highly relevant area of research for the legal, economic, and technological academic communities.

OpenChain and Software Supply Chain Security:

Investigating the depth of OpenChain adoption in the IT industry and implementation by firms, as well as its effect on actual SSC attack prevalence, could offer valuable insights.

The Intersection of AI and Open Source:

We suggest the potentials in the adaption of open-source principles towards machine learning models and training datasets – and the ethical implications behind this – as a promising topic for future research.

Bibliography

Adobe (2023): Creative Cloud for education | Educational institutions. URL <https://www.adobe.com/education.html>, retrieved October 19, 2023.

Aksulu, Altay and Wade, Michael (2010): A Comprehensive Review and Synthesis of Open Source Research. In: *J. AIS*, 11.

Al-Debei, Mutaz M and Avison, David (2010): Developing a unified framework of the business model concept. In: *European Journal of Information Systems*, 19, 3, 359–376.

Almeida, Fernando; José, Oliveira and José, Cruz (2011): Open Standards And Open Source: Enabling Interoperability. In: *International Journal of Software Engineering & Applications*, 2.

Balter, Ben (2015): 6 motivations for consuming or publishing open source software | Open-source.com. URL <https://opensource.com/life/15/12/why-open-source>, retrieved October 16, 2023.

Banon, Shay (2021): Amazon: NOT OK - why we had to change Elastic licensing, Elastic Blog. URL <https://www.elastic.co/blog/why-license-change-aws>, retrieved August 14, 2023.

Barry, Peter (2023): DuckDuckGo Usage Stats for 2023, Backlinko. URL <https://backlinko.com/duckduckgo-stats>, retrieved October 17, 2023.

Bhartiya, Swapnil (2016): What we can learn from ownCloud’s collapse | CIO. In: *CIO*.

Bitkom (2020): Open Source Monitor 2019.

Bitkom (2021): Open-Source-Monitor 2021.

Bitkom e.V (2022): Trotz Krieg und Krisen: In Deutschland fehlen 137.000 IT-Fachkräfte | Presseinformation | Bitkom e. V. URL <https://www.bitkom.org/Presse/Presseinformation/Deutschland-fehlen-137000-IT-Fachkraefte>, retrieved October 18, 2023.

Bitwarden Inc. (2023a): About Us, Bitwarden. URL <https://bitwarden.com/about/>, retrieved October 16, 2023.

Bitwarden Inc. (2023b): Bitwarden Password Manager Pricing, Bitwarden. URL <https://bitwarden.com/pricing/>, retrieved October 16, 2023.

Bitwarden Inc. (2023c): History - Status Page - Bitwarden. URL <https://status.bitwarden.com/history>, retrieved October 16, 2023.

Bitwarden Inc. (2023d): Open Source, Bitwarden. URL <https://bitwarden.com/open-source/>, retrieved October 16, 2023.

Bolívar, Luis Tomás; Camacho, Carlos and Salomon, Josh (2022): Look to the Horizon: Europe’s increased focus on funding open source research is creating new opportunities, Red Hat Research. URL <https://research.redhat.com/blog/article/look-to-the-horizon-europes-increased-focus-on-funding-open-source-research-is-creating-new-opportunities/>, retrieved October 17, 2023.

Borges, Hudson and Valente, Marco Tulio (2018): What’s in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform. In: *Journal of Systems and Software*, 146, 112–129.

Bowman, Jordan and Palmer, Taylor (2021): 2021 Design Tools Survey, UX Tools. URL <https://uxtools.co/survey/2021/>, retrieved August 30, 2023.

Bressers, Josh (2022): What is “a healthy number” of maintainers? · Issue #101 · ossf/tac, GitHub. URL <https://github.com/ossf/tac/issues/101>, retrieved October 16, 2023.

Brock, Amanda (Ed.) (2022): Open Source Law, Policy and Practice. 2nd ed., Oxford University PressOxford.

Brodkin, Jon (2021): Free software advocates seek removal of Richard Stallman and entire FSF board, Ars Technica. URL <https://arstechnica.com/tech-policy/2021/03/free-software->

advocates-seek-removal-of-richard-stallman-and-entire-fsf-board/, retrieved October 17, 2023.

BuiltWith Pty Ltd (2023): Germany: e-commerce software market share 2023, Statista. URL <https://www.statista.com/statistics/685489/e-commerce-software-provider-market-share-in-germany/>, retrieved October 18, 2023.

BuiltWith Pty Ltd: Mastodon Usage Statistics. URL <https://trends.builtwith.com/cms/Mastodon>, retrieved October 17, 2023.

Canonical Ltd. (2023): Ubuntu Pro, Ubuntu. URL <https://ubuntu.com/pro>, retrieved October 17, 2023.

Capra, Eugenio and Wasserman, Anthony I. (2008): A Framework for Evaluating Managerial Styles in Open Source Projects. In: Russo, Barbara; Damiani, Ernesto; Hissam, Scott; et al. (Eds.): *Open Source Development, Communities and Quality*. IFIP – The International Federation for Information Processing. Boston, MA, Springer US, pp. 1–14.

Carroll, Jordan (2022): The global developer community, The State of the Octoverse. URL <https://octoverse.github.com/2022/developer-community>, retrieved August 17, 2023.

Carter, Hilary; Groopman, Jessica; and The Linux Foundation: Diversity, Equity, and Inclusion in Open Source. URL <https://www.linuxfoundation.org/research/the-2021-linux-foundation-report-on-diversity-equity-and-inclusion-in-open-source>, retrieved August 11, 2023.

CBC Radio (2013): Loss Leaders: How Companies Profit By Losing Money. | CBC Radio, CBC. URL <https://www.cbc.ca/radio/undertheinfluence/loss-leaders-br-how-companies-profit-by-losing-money-1.2801812>, retrieved October 17, 2023.

Charvat, Karel; Cerba, Otakar; Ježek, Jan; et al. (2009): Open Source Business Models and Strategies.

Chesbrough, H. (2002): The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology spin-off companies. In: *Industrial and Corporate Change*, 11, 3, 529–555.

Chesbrough, Henry W. and Appleyard, Melissa M. (2007): Open Innovation and Strategy. In: *California Management Review*, 50, 1, 57–76.

Cimpanu, Catalin (2020): HackerOne's 2020 Top 10 public bug bounty programs, ZDNET. URL <https://www.zdnet.com/article/hackeron-es-2020-top-10-public-bug-bounty-programs/>, retrieved October 20, 2023.

Claburn, Thomas (2021): FSF doubles down on Richard Stallman's return: Sure, he is "troubling for some" but we need him, says org. URL https://www.theregister.com/2021/04/12/free_software_foundation_doubles_down/, retrieved October 17, 2023.

Clark, Lindsay: Tech vendors increase prices by up to 24% amid inflation. URL https://www.theregister.com/2023/06/16/tech_vendor_price_hikes/, retrieved October 16, 2023.

Clement, Mark (2019): Recent High-Profile Source Code Leaks, Stop Source Code Theft. URL <https://www.stop-source-code-theft.com/recent-high-profile-source-code-leaks/>, retrieved October 16, 2023.

cloudcommunity (2023): Cloud Service Providers Free Tier Overview.

Cobe, Patricia (2014): Feeding Gen Z, Restaurant Business. URL <https://restaurantbusinessonline.com/food/feeding-gen-z>, retrieved October 17, 2023.

Codeium Team (2023): Why did Microsoft Build VSCode? Turns out, GitHub Copilot., Codeium · Free AI Code Completion & Chat. URL <https://codeium.com/blog/why-did-microsoft-build-vscode-github-copilot>, retrieved October 17, 2023.

Creative Commons: CC0, Creative Commons. URL <https://creativecommons.org/public-domain/cc0/>, retrieved October 20, 2023.

- Dadgar, Armon (2023): HashiCorp adopts Business Source License, HashiCorp. URL <https://www.hashicorp.com/blog/hashicorp-adopts-business-source-license>, retrieved August 31, 2023.
- Dahlander, Linus and Magnusson, Mats G. (2005): Relationships between open source software companies and communities: Observations from Nordic firms. In: *Research Policy*, 34, 4, 481–493.
- Das, Ankush (2022): Google Chrome vs Chromium: What's the difference?, It's FOSS. URL <https://itsfoss.com/chrome-vs-chromium/>, retrieved October 17, 2023.
- De Blanc, Molly; Freitas, Nathan; Garrett, Matthew; et al. (2021): An open letter to remove Richard M. Stallman from all leadership positions, rms-open-letter.github.io. URL <https://rms-open-letter.github.io/>, retrieved October 17, 2023.
- DiBona, Chris; Ockman, Sam and Stone, Mark (Eds.) (1999): *Open sources: voices from the open source revolution*. 1st ed ed., Beijing ; Sebastopol, CA, O'Reilly.
- Dierking, Niklas (2023): IaC-Paukenschlag: Die Linux Foundation adoptiert den Terraform-Fork OpenTofu, c't Magazin. URL <https://www.heise.de/news/IaC-Paukenschlag-Die-Linux-Foundation-adoptiert-den-Terraform-Fork-OpenTofu-9310886.html>, retrieved October 18, 2023.
- Dinu, Flavius (2023): OpenTF is Announcing Fork of Terraform!, Spacelift. URL [https://spacelift.io/blog/\[slug\]](https://spacelift.io/blog/[slug]), retrieved October 18, 2023.
- Directorate-General for Communications Networks, Content and Technology (European Commission); Blind, Knut; Pätsch, Sivan; et al. (2021): *The impact of open source software and hardware on technological independence, competitiveness and innovation in the EU economy: final study report*. LU, Publications Office of the European Union.
- Docker Inc. (2023): What is a Container? | Docker. URL <https://www.docker.com/resources/what-container/>, retrieved October 20, 2023.
- Duparc, Estelle; Möller, Frederik; Jussen, Ilka; et al. (2022): Archetypes of open-source business models. In: *Electronic Markets*, 32, 2, 727–745.
- Engineering and Technology History Wiki (2022): Milestones:SPICE Circuit Simulation Program, 1970, ETHW. URL https://ethw.org/Milestones:SPICE_Circuit_Simulation_Program,_1970, retrieved October 16, 2023.
- European Commission (2003): 124 OJ L (European Commission) Commission Recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises (Text with EEA relevance) (notified under document number C(2003) 1422). pp. 36–41.
- European Commission (2018): COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS “Towards a common European data space.”
- European Commission (2023): The Digital Services Act package | Shaping Europe's digital future. URL <https://digital-strategy.ec.europa.eu/en/policies/digital-services-act-package>, retrieved October 17, 2023.
- European Commission: Open source software strategy. URL https://commission.europa.eu/about-european-commission/departments-and-executive-agencies/informat-ics/open-source-software-strategy_en, retrieved October 17, 2023.
- European Parliament (2022): Right to repair briefing. URL [https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/698869/EPRS_BRI\(2022\)698869_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/698869/EPRS_BRI(2022)698869_EN.pdf), retrieved October 17, 2023.
- Fairphone B.V. (2022): Impact Report - A challenge to the electronics industry, Fairphone. URL <https://www.fairphone.com/en/impact-report/>, retrieved October 17, 2023.
- FOSDEM: Home. URL <https://fosdem.org/2024/>, retrieved October 18, 2023.

FOSSA (2021a): All About Permissive Licenses - FOSSA, Dependency Heaven. URL <https://fossa.com/blog/all-about-permissive-licenses/>, retrieved October 16, 2023.

FOSSA (2021b): Building an Open Source Program Office (OSPO) - FOSSA, Dependency Heaven. URL <https://fossa.com/blog/building-open-source-program-office-ospo/>, retrieved August 11, 2023.

FOSSA (2021c): Open Source Software Licenses 101: The BSD 3-Clause License - FOSSA, Dependency Heaven. URL <https://fossa.com/blog/open-source-software-licenses-101-bsd-3-clause-license/>, retrieved October 16, 2023.

FOSSA: Commons Clause License, Commons Clause License. URL <https://commonsclause.com/>, retrieved October 20, 2023.

Framework Computer Inc. (2023): About Framework, Framework. URL <https://frame.work/de/en>, retrieved October 17, 2023.

Free Software Foundation (2021): FLOSS and FOSS - GNU Project - Free Software Foundation. URL <https://www.gnu.org/philosophy/floss-and-foss.en.html>, retrieved August 18, 2023.

Free Software Foundation (2022): What is Copyleft? - GNU Project - Free Software Foundation. URL <https://www.gnu.org/licenses/copyleft.en.html>, retrieved August 19, 2023.

Free Software Foundation (2023): What is Free Software? - GNU Project - Free Software Foundation. URL <https://www.gnu.org/philosophy/free-sw.html>, retrieved August 18, 2023.

Free Software Foundation: Proprietary Insecurity - GNU Project - Free Software Foundation. URL <https://www.gnu.org/proprietary/proprietary-insecurity.en.html>, retrieved October 16, 2023.

Gaia-X European Association for Data and Cloud AISBL: About Gaia-X - Gaia-X: A Federated Secure Data Infrastructure. URL <https://gaia-x.eu/what-is-gaia-x/about-gaia-x/>, retrieved October 17, 2023.

Ganapini, Cristina (2023): Big win for right to repair with new EU rules for batteries - but legislators must get the implementation right, Right to Repair Europe. URL <https://repair.eu/news/big-win-for-right-to-repair-with-new-eu-rules-for-batteries-but-legislators-must-get-the-implementation-right/>, retrieved October 17, 2023.

Gitea Ltd.: Compared to other Git hosting | Gitea Documentation. URL <https://docs.gitea.com/next/installation/comparison>, retrieved October 20, 2023.

GitHub Inc. (2023a): About searching on GitHub, GitHub Docs. URL <https://ghdocs-prod.azurewebsites.net/en/search-github/getting-started-with-searching-on-github/about-searching-on-github>, retrieved October 20, 2023.

GitHub Inc. (2023b): Be social, GitHub Docs. URL <https://ghdocs-prod.azurewebsites.net/en/get-started/quickstart/be-social>, retrieved October 20, 2023.

GitLab Inc. (2023a): GitLab.org / GitLab · GitLab, GitLab. URL <https://gitlab.com/gitlab-org/gitlab>, retrieved October 17, 2023.

GitLab Inc. (2023b): Our stewardship of GitLab, The GitLab Handbook. URL <https://handbook.gitlab.com/handbook/company/stewardship/>, retrieved October 17, 2023.

GitLab Inc. (2023c): Pricing. URL <https://about.gitlab.com/pricing>, retrieved October 17, 2023.

GitLab Inc. (2023d): The DevSecOps Platform. URL <https://about.gitlab.com/>, retrieved October 17, 2023.

Gonzalez-Barahona, Jesus M. (2021): A Brief History of Free, Open Source Software and Its Communities. In: Computer, 54, 02, 75–79.

Google Inc. (2020): Open source by the numbers at Google, Google Open Source Blog. URL <https://opensource.googleblog.com/2020/08/open-source-by-numbers-at-google.html>, retrieved August 11, 2023.

HackerOne: Bug Bounty Programs For Businesses | HackerOne. URL <https://www.hackerone.com/product/bug-bounty-platform>, retrieved October 20, 2023.

Haff, Gordon: The mysterious history of the MIT License | Opensource.com. URL <https://opensource.com/article/19/4/history-mit-license>, retrieved October 16, 2023.

Harutyunyan, Nikolay; Riehle, Dirk and Sathya, Gayathery: Industry Best Practices for Corporate Open Sourcing.

HashiCorp (2023): Business Source License 1.1, HashiCorp: Infrastructure enables innovation. URL <https://www.hashicorp.com/bsl>, retrieved August 31, 2023.

Hecht, Lawrence E. (2019): I Don't Git It: Tracking the Source Collaboration Market, The New Stack. URL <https://thenewstack.io/i-dont-git-it-tracking-the-source-collaboration-market/>, retrieved October 20, 2023.

heise online (2018): Bundescloud: Open-Source mit Nextcloud statt Dropbox oder Google Drive, iX Magazin. URL <https://www.heise.de/news/Bundescloud-Open-Source-mit-Nextcloud-statt-Dropbox-oder-Google-Drive-4026111.html>, retrieved August 25, 2023.

heise online (2023): GitHub ändert Home Feed – zum Ärger seiner Nutzer, Developer. URL <https://www.heise.de/news/Falscher-Feed-GitHub-veraergert-seine-Nutzer-9306448.html>, retrieved October 20, 2023.

von Hippel, Eric A. and von Krogh, Georg (2009): Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science.

Hofacker, Lars and Lanzerath, Philipp (2022): E-Commerce 2021: Zeit des Wachstums - EHI Retail Institute. EHI Retail Institute.

Homölle, Elena and Schlüter, Tim (2023): Shopware ist Marktführer in Deutschland, Shopware. URL <https://www.shopware.com/de/news/shopware-ist-marktfuehrer-in-deutschland-unter-den-shopsystemen/>, retrieved August 26, 2023.

HT Correspondent (2018): Microsoft draws flak for forcing Edge browser on Windows 10 Insiders, HT Tech. URL <https://tech.hindustantimes.com/tech/news/microsoft-draws-flak-for-forcing-edge-browser-on-windows-10-insiders-story-rvsYJCNSzt60afb5ZGo6TL.html>, retrieved October 19, 2023.

Hugging Face, Inc. (2023): Hugging Face – The AI community building the future. URL <https://huggingface.co/>, retrieved October 17, 2023.

iFixit (2023): Can Framework's Repairable Laptop Succeed? It Already Has. | iFixit News, iFixit. URL <https://www.ifixit.com/News/51695/can-frameworks-repairable-laptop-succeed-it-already-has>, retrieved October 17, 2023.

Iyer, Ram and Miller, Ron (2022): Designers underwhelmed by Adobe-Figma deal, TechCrunch. URL <https://techcrunch.com/2022/09/26/designers-underwhelmed-by-adobe-figma-deal/>, retrieved August 30, 2023.

Jina AI GmbH: About Jina AI. URL <https://jina.ai/about-us>, retrieved October 17, 2023.

Jones, Jeff (2006): Linus's Law aka “Many Eyes Make All Bugs Shallow,” Microsoft Security Blog. URL <https://www.microsoft.com/en-us/security/blog/2006/06/07/linuss-law-aka-many-eyes-make-all-bugs-shallow/>, retrieved August 11, 2023.

Karlitschek, Frank (2016): big changes: I am leaving ownCloud, Inc. today. URL <https://karlitschek.de/2016/04/big-changes-i-am-leaving-owncloud-inc-today/>, retrieved August 25, 2023.

Karlitschek, Frank (2018): Why I forked my own project and my own company.

Karlitschek, Frank and Nextcloud GmbH (2020): FOSDEM 2020: Frank Karlitschek on gpl for business. URL <https://www.youtube.com/watch?v=Xrp9iNd7oyc>, retrieved October 13, 2023.

Klein, Manfred (2022): Bundesregierung fördert Open Source mit rund 51 Millionen Euro, eGovernment. URL <https://www.egovernment.de/bundesregierung-foerdert-open-source-mit-rund-51-millionen-euro-a-1117332/>, retrieved October 17, 2023.

- Koenig, Jack (2009): Seven Open Source Business Strategies for Competitive Advantage.
- Kolf, Florian (2022): Software für E-Commerce: 100 Millionen Dollar für Shopware: Diese Brüder wollen SAP und Salesforce Konkurrenz machen. In: Handelsblatt.
- Krazit, Tom (2018): Why some open-source companies are considering a more closed approach, GeekWire. URL <https://www.geekwire.com/2018/open-source-companies-considering-closed-approach/>, retrieved October 16, 2023.
- Krill, Paul (2017): Node.js forks again, this time over a political dispute, InfoWorld. URL <https://www.infoworld.com/article/3219689/nodejs-forks-again-this-time-over-a-political-dispute.html>, retrieved October 17, 2023.
- Krishnamurthy, Sandeep (2005): An Analysis of Open Source Business Models.
- Lardinois, Frederic (2018): MongoDB switches up its open-source license, TechCrunch. URL <https://techcrunch.com/2018/10/16/mongodb-switches-up-its-open-source-license/>, retrieved October 16, 2023.
- Leigh, Roger (2014): Announcement of the Debian Fork | Devuan GNU+Linux Free Operating System, Devuan GNU+Linux. URL <https://www.devuan.org>, retrieved October 17, 2023.
- Leswing, Kif (2020): Apple's "monopoly power" over iPhone app distribution gives it outsized profits, antitrust committee says, CNBC. URL <https://www.cnbc.com/2020/10/06/house-antitrust-subcommittee-apple-has-monopoly-power.html>, retrieved October 19, 2023.
- Lim, Jamilah (2021): What's with tech companies and workplace harassment?, Tech Wire Asia. URL <https://techwireasia.com/amp/2021/08/whats-with-tech-companies-and-workplace-harassment/>, retrieved October 16, 2023.
- Lunden, Ingrid (2022a): How Mastodon is scaling amid the Twitter exodus | TechCrunch. URL <https://techcrunch.com/2022/12/23/how-mastodon-is-scaling-amid-the-twitter-exodus/>, retrieved October 17, 2023.
- Lunden, Ingrid (2022b): Penpot inks \$8M as signups for its open source spin on Figma jump 5600% after Adobe's \$20B acquisition move, TechCrunch. URL <https://techcrunch.com/2022/09/27/penpot-inks-8m-as-signups-for-its-open-source-spin-on-figma-jump-5600-after-adobes-20b-acquisition-move/>, retrieved August 30, 2023.
- Lunden, Ingrid (2022c): Penpot inks \$8M as signups for its open source spin on Figma jump 5600% after Adobe's \$20B acquisition move, TechCrunch. URL <https://techcrunch.com/2022/09/27/penpot-inks-8m-as-signups-for-its-open-source-spin-on-figma-jump-5600-after-adobes-20b-acquisition-move/>, retrieved August 30, 2023.
- Lunden, Ingrid (2023): Penpot, the open source platform for designers and their coders, draws up \$12M as users jump to 250K, TechCrunch. URL <https://techcrunch.com/2023/02/02/penpot-the-open-source-platform-for-designers-and-their-coders-draws-up-12m-as-users-jump-to-250k/>, retrieved August 30, 2023.
- Maltceva, Anna (2023): Open-Source Software (OSS) Licenses Explained | The JetBrains Platform Blog, The JetBrains Blog. URL <https://blog.jetbrains.com/platform/2023/04/open-source-software-oss-licenses-explained/>, retrieved October 16, 2023.
- MariaDB (2018): MariaDB Business Source License (BSL), MariaDB. URL <https://mariadb.com/bsl-faq-mariadb/>, retrieved August 31, 2023.
- Marsden, Christopher T. and Brown, Ian (2023): App stores, antitrust and their links to net neutrality: A review of the European policy and academic debate leading to the EU Digital Markets Act. In: Internet Policy Review, 12, 1.
- Martínez, Jeferson and Durán, Javier M. (2021): Software Supply Chain Attacks, a Threat to Global Cybersecurity: SolarWinds' Case Study. In: International Journal of Safety and Security Engineering, 11, 5, 537–545.
- Mayer, Boris (2023): JPEG XL: Die Browserhersteller sagen nein zum Bildformat - Golem.de. URL <https://www.golem.de/news/jpeg-xl-die-browserhersteller-sagen-nein-zum-bildformat-2302-171653.html>, retrieved October 20, 2023.

McLuckie, Craig: How Kubernetes came to be: A co-founder shares the story, Google Cloud Blog. URL <https://cloud.google.com/blog/products/containers-kubernetes/from-google-to-the-world-the-kubernetes-origin-story>, retrieved October 17, 2023.

Microsoft Inc.: Visual Studio Code - Code Editing. Redefined. URL <https://code.visualstudio.com/>, retrieved October 17, 2023.

Miller, Ron (2023): Terraform fork gets renamed OpenTofu, and joins Linux Foundation, TechCrunch. URL <https://techcrunch.com/2023/09/20/terraform-fork-gets-a-new-name-opentofu-and-joins-linux-foundation/>, retrieved October 18, 2023.

MongoDB Inc. (2018): Server Side Public License (SSPL), MongoDB. URL <https://www.mongodb.com/licensing/server-side-public-license>, retrieved August 31, 2023.

MongoDB Inc. (2023): Pricing, MongoDB. URL <https://www.mongodb.com/pricing>, retrieved October 16, 2023.

Morgan, Lorraine and Finnegan, Patrick (2014): Beyond free software: An exploration of the business value of strategic open source. In: *The Journal of Strategic Information Systems*, 23, 3, 226–238.

Muhammad, Hisham; Real, Lucas C. Villa and Homer, Michael (2019): Taxonomy of Package Management in Programming Languages and Operating Systems. In: *Proceedings of the 10th Workshop on Programming Languages and Operating Systems*. Huntsville ON Canada, ACM, pp. 60–66.

Nagle, Frank (2017): Learning by Contributing: Gaining Competitive Advantage Through Contribution to Crowdsourced Public Goods.

Nesbitt, Andrew and Nickolls, Benjamin (2018): Libraries.io Open Source Repository and Dependency Metadata.

Nextcloud GmbH (2016): CLA, CAA, CA etc. etc. – let's get it right - 🌍 Meta, Nextcloud community. URL <https://help.nextcloud.com/t/cla-cao-ca-etc-etc-let-s-get-it-right/588>, retrieved October 18, 2023.

Nextcloud GmbH (2023a): About Nextcloud, Nextcloud. URL <https://nextcloud.com/about/>, retrieved August 25, 2023.

Nextcloud GmbH (2023b): All apps - App Store - Nextcloud. URL <https://apps.nextcloud.com/>, retrieved October 18, 2023.

Nextcloud GmbH (2023c): Contribute to Nextcloud server or apps, Nextcloud. URL <https://nextcloud.com/contribute/>, retrieved October 20, 2023.

Nextcloud GmbH (2023d): Nextcloud Enterprise: Content collaboration software for enterprises, Nextcloud. URL <https://nextcloud.com/enterprise/>, retrieved October 20, 2023.

Nextcloud GmbH (2023e): Nextcloud Enterprise pricing, Nextcloud. URL <https://nextcloud.com/pricing/>, retrieved October 20, 2023.

Nextcloud GmbH (2023f): Nextcloud Hub - Content collaboration platform, Nextcloud. URL <https://nextcloud.com/hub/>, retrieved August 25, 2023.

Nidecki, Tomasz Andrzej (2023): Ethical hacking vs. the law – will you get arrested for a good deed?, Invicti. URL <https://www.invicti.com/blog/web-security/ethical-hacking-vs-the-law-will-you-get-arrested-for-a-good-deed/>, retrieved October 20, 2023.

Niemi, Tapio; Tuisku, Miika; Hameri, Ari-pekka; et al. (2009): Server-Based Computing Solution Based on Open Source Software. In: *Information Systems Management*, 26, 1, 77–86.

Nordhoff, Anna and Shopware (2022): Shopware presents new product portfolio, Shopware. URL <https://www.shopware.com/en/news/the-beginning-of-a-new-shopware-era-what-can-our-customers-expect/>, retrieved August 26, 2023.

NVIDIA Corporation (2023): Adobe and NVIDIA Partner to Unlock the Power of Generative AI, NVIDIA Newsroom. URL <http://nvidianews.nvidia.com/news/adobe-and-nvidia-partner-to-unlock-the-power-of-generative-ai>, retrieved October 19, 2023.

Oberhauser, Jan (2021): Fair-code: the future for sustainable open source alternatives 🍷, n8n Blog. URL <https://blog.n8n.io/fair-code-for-sustainable-open-source-alternatives/>, retrieved October 18, 2023.

Ogden, Cody (2024): Killed by Google, Killed by Google. URL <https://killedbygoogle.com>, retrieved October 16, 2023.

Okoli, Chitu and Nguyen, An (2015): Business Models for Free and Open Source Software: Insights from a Delphi Study.

Open Source Business Alliance, Bundesverband für digitale Souveränität e.V. (2021): Gaia-X beweist Unabhängigkeit von großen US-Tech-Anbietern und entscheidet sich für EU-basierte Kollaborationsplattform, OSBA – Open Source Business Alliance. URL <https://osb-alliance.de/news/gaia-x-entscheidet-sich-fuer-eu-basierte-kollaborationsplattform>, retrieved August 25, 2023.

Open Source Contributor Index: OSCI – Open Source Contributor Index. URL <https://opensourceindex.io/>, retrieved August 11, 2023.

Open Source Initiative (2006a): The License Review process, Open Source Initiative. URL <https://opensource.org/licenses/review-process/>, retrieved August 19, 2023.

Open Source Initiative (2006b): The Open Source Definition, Open Source Initiative. URL <https://opensource.org/osd/>, retrieved August 14, 2023.

Open Source Initiative (2007): The Open Source Definition (Annotated), Open Source Initiative. URL <https://opensource.org/definition-annotated/>, retrieved August 19, 2023.

Open Source Initiative (2018): History of the OSI, Open Source Initiative. URL <https://opensource.org/history/>, retrieved August 14, 2023.

Open Source Initiative (2022): Licenses, Open Source Initiative. URL <https://opensource.org/licenses/>, retrieved August 19, 2023.

OpenTofu (2023a): FAQ | OpenTofu. URL <https://opentofu.org/faq/>, retrieved October 18, 2023.

OpenTofu (2023b): OpenTofu Announces Fork of Terraform | OpenTofu. URL <https://opentofu.org/blog/opentofu-announces-fork-of-terraform/>, retrieved October 18, 2023.

OpenTofu (2023c): opentofu/opentofu: OpenTofu lets you declaratively manage your cloud infrastructure. URL <https://github.com/opentofu/opentofu>, retrieved October 18, 2023.

O'Reilly Media Inc.: Open Source Software Conference - O'Reilly Media. URL <https://www.oreilly.com/conferences/oscon.html>, retrieved October 18, 2023.

ownCloud GmbH (2016): ownCloud Secures Financing and Expands its Management Team - ownCloud. URL <https://owncloud.com/news/owncloud-secures-financing-expands-management-team/>, retrieved August 25, 2023.

Pasero, Benjamin (2021): VS Code — The Story and Technology Behind One of the World's Most Popular Desktop Apps for Developers. URL <https://www.git-tower.com/blog/developing-for-the-desktop-vscode/>, retrieved October 17, 2023.

Penpot (2023a): Penpot - About. URL <https://penpot.app/about>, retrieved August 30, 2023.

Penpot (2023b): Penpot - Features. URL <https://penpot.app/features>, retrieved August 30, 2023.

Penpot (2023c): Penpot Community, Penpot Community. URL <https://community.penpot.app/>, retrieved October 20, 2023.

Penpot (2023d): Penpot Fest Barcelona June 28-30 Building Freedom for Design, registration now open!, Penpot Fest. URL <https://penpotfest.org/>, retrieved October 20, 2023.

Penpot (2023e): penpot/penpot.

phendrenad2 (2021): Github really needs to allow people to disable the "Pull Request" tab on repos, ... | Hacker News. URL <https://news.ycombinator.com/item?id=25940799>, retrieved October 16, 2023.

Pike, Paoli: WHITE PAPER ON THE SEARCH ENGINE MARKET.

Poortvliet, Jos (2017): Nextcloud box updates: Raspberry Pi 3, Nextcloud 12 and looking for a new provider, Nextcloud. URL <https://nextcloud.com/blog/nextcloud-box-updates-raspberry-pi-3-nextcloud-12-and-looking-for-a-new-provider/>, retrieved October 20, 2023.

Preston-Werner, Tom (2011): Open Source (Almost) Everything. URL <https://tom.preston-werner.com/2011/11/22/open-source-everything.html>, retrieved October 16, 2023.

PricewaterhouseCoopers (2021): Open Source Software Study 2021 – Recent Trends and Developments, PwC. URL <https://www.pwc.de/en/digitale-transformation/open-source-software-study-2021-recent-trends-and-developments.html>, retrieved August 17, 2023.

Prototype Fund: Open Source als Geschäftsmodell: Frank Karlitschek von Nextcloud.

Rajala, Risto; Nissilä, Jussi and Westerlund, Mika (2006): Determinants of OSS revenue model choices. In: pp. 1839–1850.

Ramač, Robert; Mandić, Vladimir; Taušan, Nebojša; et al. (2022): Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry. In: *Journal of Systems and Software*, 184, 111114.

Red Hat Inc. (2022): The State of Enterprise Open Source 2022. URL <https://www.redhat.com/en/enterprise-open-source-report/2022>, retrieved August 1, 2023.

Red Hat Inc. (2023a): Red Hat Enterprise Linux operating system. URL <https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux>, retrieved October 17, 2023.

Red Hat Inc. (2023b): What is Infrastructure as Code (IaC)? URL <https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac>, retrieved October 20, 2023.

Riehle, Dirk (2012): The single-vendor commercial open course business model. In: *Inf. Syst. E-Business Management*, 10, 5–17.

Riehle, Dirk (2019): The Innovations of Open Source. In: *Computer*, 52, 59–63.

Robles, Gregorio and González-Barahona, Jesús M. (2012): A Comprehensive Study of Software Forks: Dates, Reasons and Outcomes. In: Hammouda, Imed; Lundell, Björn; Mikkonen, Tommi; et al. (Eds.): *Open Source Systems: Long-Term Sustainability*. IFIP Advances in Information and Communication Technology. Berlin, Heidelberg, Springer, pp. 1–14.

Ruiz-Múzquiz, Pablo (2022): Penpot: Our Time Has Come - Events and Announcements, Penpot Community. URL <https://community.penpot.app/t/penpot-our-time-has-come/1563>, retrieved August 30, 2023.

Ruiz-Múzquiz, Pablo (2023a): 5 open source AI challenges with Penpot! - Events and Announcements, Penpot Community. URL <https://community.penpot.app/t/5-open-source-ai-challenges-with-penpot/3327/3>, retrieved October 20, 2023.

Ruiz-Múzquiz, Pablo (2023b): FAQ: What are the plans to charge for Penpot at some point? - FAQ, Penpot Community. URL <https://community.penpot.app/t/faq-what-are-the-plans-to-charge-for-penpot-at-some-point/2561>, retrieved October 20, 2023.

Runa Capital: The fastest-growing open-source startups in Q1 2023, Runa Capital. URL <https://runacap.com/ross-index/q1-2023/>, retrieved October 19, 2023.

Scarfone, Karen; Jansen, Wayne and Tracy, Miles (2008): Guide to General Server Security. National Institute of Standards and Technology.

Scherschel, Fabian A. (2023): Modern Solution: Jetzt doch Strafverfahren gegen Sicherheitsforscher, heise online. URL <https://www.heise.de/news/Modern-Solution-Jetzt-doch-Hackerparagraf-Verfahren-gegen-Sicherheitsforscher-9246117.html>, retrieved October 20, 2023.

Scheuer, Stephan and Kerkmann, Christof (2019): Cloud-Dienste: EU-Staaten setzen auf deutsches Start-up Nextcloud. URL <https://www.handelsblatt.com/technik/it-internet/cloud->

dienste-eu-staaten-setzen-auf-deutsches-start-up-nextcloud/24942352.html, retrieved August 25, 2023.

Schmidt, Julia (2022): Qt rethinks commercial licensing, so you can distribute apps after your license has expired • DEVCLASS, DEVCLASS. URL <https://devclass.com/2022/02/02/qt-company-commercial-license-change/>, retrieved October 20, 2023.

Schneider, Mikaela (2023): Nextcloud Ethical AI Rating, Nextcloud. URL <https://nextcloud.com/blog/nextcloud-ethical-ai-rating/>, retrieved October 18, 2023.

Senz, Kristen and Nagle, Frank (2018): The Hidden Benefit of Giving Back to Open Source Software, HBS Working Knowledge. URL <http://hbswk.hbs.edu/item/the-hidden-benefit-of-giving-back-to-open-source-software>, retrieved August 11, 2023.

Shafranovich, Yakov and Foudil, Ed: security.txt, security.txt. URL <https://securitytxt.org/>, retrieved October 20, 2023.

Shah, Sonali K. and Nagle, Frank (2020): Why Do User Communities Matter for Strategy? In: Strategic Management Review, 1, 2, 305–353.

Sharma, Chinmayi (2022): Tragedy of the Digital Commons. In: SSRN Electronic Journal.

Shopware AG (2023a): Plans & Pricing, Shopware. URL <https://www.shopware.com/en/pricing/>, retrieved August 26, 2023.

Shopware AG (2023b): Shopware. URL <https://github.com/orgs/shopware/repositories>, retrieved October 20, 2023.

Shopware AG (2023c): Shopware.

shopware AG (2023): Shopware Community Edition, Shopware. URL <https://www.shopware.com/en/community/community-edition/>, retrieved October 20, 2023.

Shopware AG (2023d): The story behind shopware AG, Shopware. URL <https://www.shopware.com/en/company/story/>, retrieved August 26, 2023.

Slack, Quinn (2023a): Missing Apache license file · Issue #53528 · sourcegraph/sourcegraph, GitHub. URL <https://github.com/sourcegraph/sourcegraph/issues/53528>, retrieved October 17, 2023.

Slack, Quinn (2023b): Sourcegraph is no longer open source | Hacker News. URL <https://news.ycombinator.com/item?id=36584656>, retrieved October 17, 2023.

Smith, McCoy (2019): Presentation: “Extreme” Copyleft: Boon, or Bane? In: Brussels, Belgium, Software Freedom Conservancy.

Spearrin, Kyle; Augendre, Gabriele; Taikon, Kutsuro; et al. (2018): Open-Source: Can Bitwarden be exploited? - Ask the Community / Password Manager, Bitwarden Community Forums. URL <https://community.bitwarden.com/t/open-source-can-bitwarden-be-exploited/1264>, retrieved October 16, 2023.

Stallman, Richard (2008): Richard Stallman on the political nature of the Free software movement, emacs-devel mailing list. URL <https://lists.gnu.org/archive/html/emacs-devel/2008-03/msg00635.html>, retrieved October 16, 2023.

Stallman, Richard (2022): Why Open Source Misses the Point of Free Software - GNU Project - Free Software Foundation. URL <https://www.gnu.org/philosophy/open-source-misses-the-point.html>, retrieved August 1, 2023.

Statistics and Data: Top Companies Contributing to Open Source - 2011/2020 - YouTube. URL <https://www.youtube.com/watch?v=iaF8MV2uKzs>, retrieved October 20, 2023.

Stiefel, Michael; Meeker, Heather J. and Dix, Paul: Will Cloud Computing Kill Open Source Development?, InfoQ. URL <https://www.infoq.com/articles/will-cloud-computing-kill-open-source/>, retrieved October 16, 2023.

Swire, Peter (2004): A Model for When Disclosure Helps Security: What Is Different About Computer and Network Security?

Synopsys (2022): 2022 OSSRA Report.

Taft, Darryl K. (2021a): Microsoft VS Code: Winning developer mindshare | TechTarget, Software Quality. URL <https://www.techtarget.com/searchsoftwarequality/news/252496429/Microsoft-VS-Code-Winning-developer-mindshare>, retrieved October 17, 2023.

Taft, Daryll K. (2021b): Survey: Open source skills more valued than proprietary ones | TechTarget, App Architecture. URL <https://www.techtarget.com/searchapparchitecture/news/252496117/Survey-Open-source-skills-more-valued-than-proprietary-ones>, retrieved October 19, 2023.

Taiga: Taiga: Pricing. URL <https://taiga.io/pricing-cloud>, retrieved October 20, 2023.

TeX Users Group: History of TeX - TeX Users Group. URL <https://www.tug.org/whatis.html>, retrieved October 16, 2023.

The Kubernetes Authors (2023): Creating Highly Available Clusters with kubeadm, Kubernetes. URL <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/high-availability/>, retrieved October 16, 2023.

The Linux Foundation: Winding Down an Open Source Project. URL <https://www.linuxfoundation.org/resources/open-source-guides/winding-down-an-open-source-project>, retrieved August 1, 2023.

The Linux Foundation and The Harvard Laboratory for Innovation Science (2020): Report on the 2020 FOSS Contributor Survey. URL <https://www.linuxfoundation.org/resources/publications/foss-contributor-2020>, retrieved August 11, 2023.

The Open Group: The UNIX System -- History and Timeline -- UNIX History. URL https://unix.org/what_is_unix/history_timeline.html, retrieved October 16, 2023.

The OpenChain Project and The Linux Foundation (2023): License-Compliance-Specification/2.1/en/openchainspec-2.1.pdf at master · OpenChain-Project/License-Compliance-Specification. URL <https://github.com/OpenChain-Project/License-Compliance-Specification/blob/master/2.1/en/openchainspec-2.1.pdf>, retrieved September 9, 2023.

Tremmel, Moritz (2022): Nach Datenleck bei Modern Solution: Vorwürfe statt Entschuldigung - Golem.de. URL <https://www.golem.de/news/nach-datenleck-bei-modern-solution-vorwuerfe-statt-entschuldigung-2202-162762.html>, retrieved October 20, 2023.

Tsukayama, Hayley (2023): Right to Repair Advocates Have Had Good Victories. We Have To Keep Fighting., Electronic Frontier Foundation. URL <https://www.eff.org/deeplinks/2023/01/right-repair-advocates-have-had-good-victories-we-have-keep-fighting>, retrieved October 17, 2023.

U. S. Government Accountability Office (2022): Cybersecurity: Federal Response to SolarWinds and Microsoft Exchange Incidents | U.S. GAO. URL <https://www.gao.gov/products/gao-22-104746>, retrieved October 20, 2023.

United States Department of Defense (2021): Open Source Software FAQ. URL <https://dodcio.defense.gov/Open-Source-Software-FAQ/>, retrieved October 16, 2023.

UpGuard (2023): The 72 Biggest Data Breaches of All Time [Updated 2023] | UpGuard. URL <https://www.upguard.com/blog/biggest-data-breaches>, retrieved October 16, 2023.

Varghese, Sam (2021): iTWire - Pro-Stallman group issues open letter, wants him to stay on FSF board. URL <https://itwire.com/business-it-news/open-source/pro-stallman-group-issues-open-letter,-wants-him-to-stay-on-fsf-board.html>, retrieved October 17, 2023.

Vaughan-Nichols, Steven (2016): OwnCloud founder forks popular open-source cloud. In: ZDNET.

Vaughan-Nichols, Steven (2019): MongoDB “open-source” Server Side Public License rejected, ZDNET. URL <https://www.zdnet.com/article/mongodb-open-source-server-side-public-license-rejected/>, retrieved October 16, 2023.

Vaughan-Nichols, Steven (2022): GitHub vs GitLab: Which program should you go with?, ZDNET. URL <https://www.zdnet.com/article/github-vs-gitlab-the-key-differences/>, retrieved October 20, 2023.

Weatherbed, Jess (2022): Here's another example of the frustrating "Adobe upsell," The Verge. URL <https://www.theverge.com/2022/9/20/23362669/reason-why-people-hate-adobe-photoshop-airdrop-removed-file-sharing>, retrieved October 19, 2023.

Weatherbed, Jess (2023): Adobe Express rolls out new generative AI features, The Verge. URL <https://www.theverge.com/2023/8/16/23834146/adobe-express-firefly-generative-ai-release-design-app>, retrieved October 19, 2023.

webfellows UG (2022): Das neue Preis- und Lizenz-Model von Shopware 2022, Shopware Full Service Agentur. URL <https://www.webfellows.eu/shopware-preise-2022/>, retrieved October 19, 2023.

Wu, Chorng-Guang; Gerlach, James H. and Young, Clifford E. (2007): An empirical analysis of open source software developers' motivations and continuance intentions. In: *Information & Management*, 44, 3, 253–262.

Yin, Robert K. (2018): *Case study research and applications: design and methods*. Sixth edition ed., Los Angeles, SAGE.

Young, Jennifer (2018): KeePass: Why the Free, Open-Source Password Manager is Our Top Pick for Protecting Your Login Information, HostingAdvice.com. URL <https://www.hostingadvice.com/blog/keepass-why-the-free-open-source-password-manager-is-our-top-pick/>, retrieved October 16, 2023.

Zaitsev, Peter (2023): The Future of Open Source is Polarized, Percona Database Performance Blog. URL <https://www.percona.com/blog/the-future-of-open-source-is-polarized/>, retrieved October 18, 2023.

Appendix 1: Code Hosting Platform Comparison Table

Features Across Platforms							
Features	Code review	Yes	Yes	Yes	Yes	Yes	Yes
	Bug tracking	Yes	Yes	Yes	Yes	Yes	Yes
	Web hosting	Yes	Yes	No	Yes	Yes	Yes
	Wiki	Yes	Yes	Yes	Yes	Yes	Yes
	Mailing list	No	No	No	Yes	Yes	No
	Forum	Yes	No	No	Yes	Yes	No
	Personal repository	Yes	Yes	Yes	Yes	Yes	Yes
	Private repository	Yes	Yes	Yes	Yes	Yes	Yes
	Build system	Yes	Yes	Yes	Yes	No	Yes
	Team	Yes	Yes	Yes	Yes	Yes	Yes
	Release binaries	Yes	Yes	Yes	Yes	Yes	No
	Self-hosting	Commercially	Yes	Yes	Commercially	Yes	Commercially
	Project management	Yes	Yes	Yes	Yes (Azure Boards)	No	Jira Integration
	Issue tracking	Yes	Yes	Yes	Yes (Azure Boards)	No	Yes
	Telemetry-free	No	No	Yes	No	No	No
	Analytics	Yes	Yes	Yes	Internal only	Yes	Internal only
	Package Registry	Yes	Yes	Yes	No	No	No
	Repo Sync	Yes	Yes	Yes	No	No	No
	Global search	Yes	Yes	No	No	Yes	No
	LFS	Yes	Yes	Yes	Yes (HTTPS only)	No	Yes
	LDAP/SAML/OIDC/OAuth 2.0	Yes/Yes/Yes/Yes	Yes/Yes/Yes/Yes	Yes/No/Yes/Yes	Yes/Yes/Yes/Yes	No/No/No/No	Yes/Yes/Yes/Yes
	Social Features	Yes	Yes	Yes	No	Yes	No
		GitHub	GitLab	Gitea	Azure DevOps	SourceForge	Bitbucket

Platforms

Appendix 2: Open-Source License Comparison Table

Attributes	LGPL-3.0	AGPL-3.0	GPL-3.0	Unlicense	GPL-2.0	Apache-2.0	BSD-2-Clause	BSD-3-Clause	MIT	CC0-1.0
FSF approved	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
OSI approved	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
GPLv3 compatible	Yes	Yes	Yes	Yes	No	GPLv3	Yes	Yes	Yes	Yes
Copyleft	Yes	Yes	Yes	No	Yes	No	No	No	No	No
Linking exception	Yes	GPLv3	GPL	Yes	GPL	Yes	Yes	Yes	Yes	Public Domain
Debian approved	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Partial
Fedora approved	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Patent grant	Yes	Yes	Yes		Yes	Yes	Manually	Manually	Manually	No
Trademark grant	Yes	Yes	Yes		Yes	No	Manually	Manually	Manually	No
Private use	Yes	Yes	Yes	Public Domain	Yes	Yes	Yes	Yes	Yes	Public Domain
Sublicensing	Copylefted	Copylefted	Copylefted	Public Domain	Copylefted	Permissive	Permissive	Permissive	Permissive	Public Domain
Modification	Copylefted	Copylefted	Copylefted	Public Domain	Copylefted	Permissive	Permissive	Permissive	Permissive	Public Domain
Distribution	Copylefted	Copylefted	Copylefted	Public Domain	Copylefted	Permissive	Permissive	Permissive	Permissive	Public Domain
Linking	Restricted	GPLv3 only	GPLv3	Public Domain	GPLv2	Permissive	Permissive	Permissive	Permissive	Public Domain

Annotation:

- **Approvals by organizations:** Whether the License is approved for use by the respective organization in their software, releases, distributions, etc.
- **GPL/GPLv3/GPLv2:** This denotes that software can be combined with software under the GNU General Public License of the respective version.

- **Linking exception:** This allows proprietary software to link to a library without being subject to the library's open-source license.
- **Patent grant:** This clause indicates that contributors provide an implicit grant of their patent rights when contributing to a project.
- **Private use:** Indicates whether modifications of the software can be kept private.
- **Sublicensing:** Describes if rights and obligations of the license can be transferred via a sublicense.
- **Modification & Distribution:** Describes the rights and limitations for users to modify and distribute the software.

Appendix 3: Case Study Questionnaire Questions

Preface

Hello, and thank you for participating in our case study on 'Monetization & Growth Strategies for Open-Source Software Companies.'

Your organization was one of three which we selected as exemplary candidates for our research. 🌿

We would be delighted to gather your thoughts on the following questions and incorporate your experiences into our analysis.

Please refer to this document for information about how your response will be handled and which information about you will be disclosed in the final document:

The questionnaire consists of 11 questions and will take approximately 20 minutes to complete.

Of course, this depends on the length of your answers. 😊

There are no limits on the length or content of your responses; feel free to give answers that are as brief or extensive as you deem appropriate.

You may skip any questions you do not want to answer.

Q0: Your Name and Occupation

1. Please provide your name and current occupation.

This will not be shared in the publicly accessible variant of the thesis. If you omit this question, your response will be recorded as an anonymous response.

Q1: Your experience

1. Please give an overview of your professional experience with Open-Source software.

Q2: Growing an Open-Source project - Code hosting platforms

1. In your experience, did your organization's choice of a code hosting platform (GitHub, GitLab, etc.) have a significant impact on the growth of your userbase or community?
2. Was this impact positive or negative?
3. What features does the platform offer that were the reason for your organization to use it?

Q3: Open-Source Licensing

1. How did you (or your organization) decide on an open-source license for your product(s)?
2. How would you describe this choice's impact on your community of users and contributors?
3. How would you describe this choice's impact on your business strategy?

Q4: Open-Source Values and Monetization

1. Have there been instances where you (or your organization) had to compromise between the spirit of openness and the financial realities of running a company offering open-source software?
2. If so, how did you reconcile the situation?

Q5: Pivotal Strategies

1. When thinking about your organization's history, do any specific measures (management, technological, legal) come to mind that had a significant impact on your project's growth?

Q6: Open-Source Business Models

There are several typical business models (open core, dual licensing, etc.) commonly observed in open-source companies.

1. In which business model category would you place your organization?

2. How did you (or your organization) arrive at your current business model?
3. Were other business models considered or tried?
4. Were there situations where this model clearly hindered the growth of your community, market share, revenue, or profit?

Q7: Open-Source Monetization

There are several typical revenue mechanisms (subscriptions, pay-per-use, advertising, donations, etc.) commonly observed in open-source companies.

1. In which monetization category would you place your organization?
2. How did you (or your organization) arrive at your current monetization mechanism?
3. Were other mechanisms considered or tried?
4. Were there situations where your monetization mechanism clearly hindered the growth of your community, market share, revenue, or profit?

Q8: The perception of Open-Source Software in the industry

Based on your experience, how would you describe the attitude towards open-source software in the IT industry when compared to five years ago?

- Much more favorable
- Slightly more favorable
- No change
- Slightly less favorable
- Much less favorable

Q9: The perception of Open-Source Software in the public eye

Based on your experience, how would you describe the awareness of the general public - not just IT professionals - of the existence and importance of open-source software?

- Much more aware
- Slightly more aware
- No change
- Slightly less aware
- Much less aware

Q10: Open-Source vs. Source-available

In recent years, notable open-source projects transitioned from their previous open-source licensing model towards a 'source-available' model which limits the scope in which their product can be commercially exploited - for example, the SSPL or BSL licenses.

The projects cite reasons such as pressure from large firms offering their products as services which leads to an erosion of their own revenue streams.

The reactions from community members and industry professionals have been mixed, but sometimes very vocal.

Supporters claim that this move is the only way to give the developers of these products a chance to stay financially afloat.

Opponents criticize the move away from OSI-approved licenses as a way to "save open-source by killing it" with some even forking older versions and providing updates under an OSI-approved license.

1. Please offer your thoughts on this debate.
2. Do you foresee an improvement in this issue in the coming years?

3. Are you in support of 'source-available' licenses, or are you critical of them?
4. Do you perceive any alternatives to 'source-available' licenses for open-source companies being crowded out by large service providers?

Q11: Advice

1. What advice would you give to someone willing to create a company around their open-source project?
2. What advice would you give to the decision-makers of a company which is selling proprietary, closed-source software, but considering open-sourcing their product?

Space for free-form entry

You can use this entry to provide any input on our research topic which was not covered by one of the pre-written questions.

Thank you so much for sharing your thoughts!

Contact information

If you wish to receive a copy of the thesis when it is submitted, you may provide an e-mail address or other means of contact here. We will provide a final copy to you and delete this information afterwards.

This information will not be used or shared in any other way, no matter if you chose to submit your answers anonymously or not.

Appendix 4: Questionnaire Response from P1 & P2

Q0: Your Name and Occupation

- Redacted in public release -

Q1: Your experience

5 years

Q2: Growing an Open-Source project - Code hosting platforms

At kaleidos we use GitHub. We see a valuable impact as our user can easily find us and our code because it is where the community is. We didn't consider any feature relevant. However, the GitHub stars and forks is a way for us to measure the impact of our project in the developer world.

Q3: Open-Source Licensing

Kaleidos Open Source started in Spain back in 2011 as a software consultancy that believed strongly in open source. We decided early on that if you are a user of open-source software, you also need to give back to the open-source community. This felt like the ethical thing to do, so we agreed early on that we would only sign up clients that would be willing to let our work be shared back with the community. This was a tough business decision - it was the middle of the Spanish financial crisis.

To be open source, it's our very personal choice to make sure that we create tools that inject more freedom into the system. Open Source means pursuing a fairer society, where opportunities are more evenly distributed. Software Technology has the unique advantage, compared to other industries and intellectual property, of having almost zero cost to replicate itself, thus providing a wonderful chance to massively distribute the tools for a more digitally sovereign society.

Besides the pure license aspect of it and its legal framework, Open Source fosters more engaging communities where the lines between user and contributor are often blurred.

In particular, we chose the very respected Mozilla Public License 2.0 because it made software delivered through the web or a service subject to the same rules as software that runs natively on your operating system.

For more information on why kaleidos believe on open source check blog post here: <https://blog.kaleidos.net/Why-Open-Source/>

In 2020 Kaleidos decided to focus on their two projects Penpot and Taiga. Building two open-source tools is a massive undertaking. We sought public funding in Spain and were granted enough to kickstart a small team to begin working full-time on it. We were fortunate to be able to raise our first external funding from several well-respected business leaders in Spain and financed the rest of the capital from employee savings. Furthermore, we announced our Penpot Alpha and won Product of the Day at Product Hunt and received tons of love on sites like HN. We knew we had struck a chord with our community and went on to raise our Seed round with Athos Capital and CDTI Invierte. Our investors always shared alignment with us on the open source nature of the project and by November 2021 we were able to move the Penpot project to the Beta stage and experienced a new surge in adoption and love.

Our community was growing fast and we had a long list of large enterprises such as Google, Microsoft, Red Hat, Tencent, Bytedance and Mozilla that were all starting to use Penpot internally as well as communities like Blender. The project was growing in ambition and we decided we needed to take on another investment to support the project, and set out to find an investor who shared our same vision. Most investors will love what you have achieved but also feel strongly about creating immediate commercial results. We were excited to learn that Jon, Sudip and Dan from Decibel were a different group of investors. Decibel lead

this \$8M series A round, the biggest series A for a Spanish open-source company to date.

Q4: Open-Source Values and Monetization

I would say no. However, we have received concerned regarding taking VC money from the community after we announced the series A round. We try to be as transparent as we can with all this issues and try to explain our decisions.

You can find more about this discussing here: <https://community.penpot.app/t/penpot-our-time-has-come/1563/15>

Q5: Pivotal Strategies

One big pivotal Strategy was changing from treating Taiga and Penpot as side project to full focus on them in 2020. It undoubtedly made both projects bigger and with more influence in the sector.

Q6: Open-Source Business Models

Most likely we will follow an open core model where the paid tiers target medium to big organizations in need of specific features that are not needed by power users, who will enjoy Penpot Free Forever. Other options like marketplace fees for paid templates and libraries could also be an option. We expect our first paid tier to be ready by the end of 2024. Right now, this is not our focus at all due to the series A.

Q7: Open-Source Monetization

Right now, Penpot is free and Taiga is support subscription base .

Our previous experience with Taiga was a great learning experience on how different monetization strategies can work for an open-source product. SaaS-only per-seat freemium pricing

doesn't really work at scale. Your self-host option which we believe has to be ridiculously easy to deploy) will quickly cannibalize your SaaS user base. This is particularly true for team productivity tools where there is a clear incentive to spin out your own instance instead of going SaaS. The good news is that if you build a truly robust platform, you can have tens of thousands of active deployments and yet have a pretty silent tech support channel.

This is why we strongly believe our pricing will cover self-host and SaaS options at the same time, so we really don't care which Penpot deployment option you prefer. For this we have two simple rules:

Everything power users care about will always be open source and therefore free for ever. A power user is typically a designer or a developer that uses Penpot as one of their key productivity tools to deliver value to a project.

Anything that was once open source will remain open source forever.

So what to charge for and to whom? We're looking at enterprise features needed by medium to big organizations to ensure legal or security compliance (think of a company-wide enforced 2FA 2). Also, features that bring value to senior management (think of multi-team views/controls/reporting). This revenue is funneled into ensuring we keep innovation at both the open source level and the premium level.

Now, the obvious challenge here is that if you go for a skinny open-core model where 99% of your product is open-source, you need a huge footprint to be successful at a business level and with that sustainability. You need to give away a lot before considering asking for some money. That is the reason why our focus can't be right now on "premium features" when there's still so much at stake in terms of the open-source product. We chose investors that are 100% aligned with this and we love how they publicly have shared these same ideas and values!

Q8: The perception of Open-Source Software in the industry

Slightly more favorable

Q9: The perception of Open-Source Software in the public eye

Much more aware

Q10: Open-Source vs. Source-available

Source-available in the past has been synonym of "corporate hijacking" because it came from a very different standpoint. See Microsoft in the Steve Balmer era. Now, it is very different, because it comes from the open source industry. Our initial reaction is of empathy since we don't think they do this willingly but out of the realisation that, for them, the game theory model is unfairly broken. Of course, we don't want to see more of that but on an infrastructure-dominated open source arena, this is an inherent weakness today "thanks" to the commoditisation of cloud. We need to come up with OSI-approved licenses that are step forward. I don't think we're equipped with the right legal toolkit at the moment, it's aging bad. We're critical of them but understand that it might be a desperate measure. No, and that's the issue. We need something that feels 2023 or even 2030 and that requires a ton of thought and work. We want to work on that.

Q11: Advice

Be aware of where do you stand in the overarching value chain and how easily would be for a third party to "intermediate" you as a proxy to the end user. In other words, who "owns" the relationship with your userbase (userbase could also be developers). This is relevant if you're trying to build a company. Also, how much value are you going to create globally. You will only capture a tiny percentage of it. Be careful with niche products. For 9 out of 10 cases, it's already too late. Be sure to understand the different power dynamics and conflict of interests that come from nurturing a community of users and contributors

Space for free-form entry

The two last questions were answered by [P2 - name redacted in public release] who has more knowledge in the area

Usage of Responses

I agree

Appendix 5: Questionnaire Response from P3

Q1: Your experience

5 years working in OSS teams as engineer + 3.5 as project manager

Q2: Growing an Open-Source project - Code hosting platforms

1: GitHub definitely played a role in driving engagement with the product. We actively promote contributions there and try to make it as easy as possible for ppl to join in.

2: Positive.

3: In terms of SCM platforms, GitHub stands out, of course due to its market share and because it's so widely used, but also because it's the only platform that really makes community building at this scale possible. There are other great platforms for SCM with their own strengths, but GitHub's specialty is the way it can really engage developers and users.

Q3: Open-Source Licensing

1 & 2: I was not there for this phase of the project but I can speak on licenses in general. From my view, licensing is most important for businesses that want to use your product or integrate the code in some way. The end user doesn't care about the license, I would even guess that most don't even look at it, except for when there is a good reason to.

3: N/a

Q4: Open-Source Values and Monetization

1: Not that I am directly aware. Although I have seen this issue from both sides in the past while working on other projects. When you ask about "Financial Realities" the recent debate around BSL comes to mind. I would say that this is a prime example of a necessary compromise between the "spirit of openness" and staying

afloat financially. Also, there is debate to be had around models like open core etc. because they have sometimes a lot of proprietary (not open) components to them. For me, it really depends on the implementations, but there are lots of great people in the industry who have strong opinions on either side of the topic.

2: N/a

Q5: Pivotal Strategies

1: Once again, I cannot speak for the entire history of the project, but I can say that focusing on public sector and education are strong areas of opportunity for open source. This is where politics might come into the mix, but I believe that public investment should go into free and open software, and not into black boxes developed by tech giants with dubious motives. But this is absolutely my personal opinion, even though it seems many developers agree with it.

Q6: Open-Source Business Models

Participant did not provide an answer.

Q7: Open-Source Monetization

Participant did not provide an answer.

Q8: The perception of Open-Source Software in the industry

Slightly more favorable

Q9: The perception of Open-Source Software in the public eye

Slightly more aware

Q10: Open-Source vs. Source-available

This is in my opinion a great question. The degree of risk from the big providers like AWS, GCP and so on depends on how you

make your money a lot. The more you rely on a relationship-based business model, the more risk there is from these managed service providers and hyperscalers to come in as a third party and kind of "hijack" the relationship between you and your customers without really giving anything back to the company that actually develops the software. It's a slippery slope for sure, and there are great debates to be had about the best approach to prevent this. Luckily for many companies that offer support and consulting services, those are things that you can not as easily replicate as, for example, managed hosting. Adopting other revenues streams like this can be an alternative to a licensing change. Once again, this is my personal opinion, but I think a change away from open source licenses should be a last resort that is, in my opinion, often not necessary and can instead be tackled with a change in the business model first and foremost.

Q11: Advice

1: Be as open and transparent as possible, try to know your customers and users before you make decisions on business models. Hiring devs that have open source experience can be a great way to achieve this. I know, easier said than done!

The second point I would like to raise is to put your product - your software - at the center of your attention, especially early on. Marketing, sales, hype, etc. is good to have for sure, but great product and great features trumps everything else, especially when you're planning in the long term and not just going for short term attraction.

2: Go for it! There is a massive push for open source, it has a lot of momentum right now. A lot of engineers favor open source companies and the open spirit. So it can be an opportunity for you to recruit great new talent, which is of course very sought after these days.

Usage of Responses

I agree