Discussion Paper Series – CRC TR 224

Discussion Paper No. 585
Project B 05

Free and Open-Source Software: Coordination and Competition

Robin Ng[1]

August 2024

[1]University of Mannheim, Email: hello@robinng.com

Support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through CRC TR 224 is gratefully acknowledged.

Collaborative Research Center Transregio 224 - www.crctr224.de
Rheinische Friedrich-Wilhelms-Universität Bonn - Universität Mannheim

Free and Open-Source Software: Coordination and Competition

Robin Ng†

29th July 2024

Abstract

Free and Open-Source Software (FOSS) are developed by a community of developers led by a coordinator. Coordinators balance the following trade-off: (i) more developers improve FOSS quality—a positive vertical differentiation effect; (ii) more developers lead to more diverse views, driving FOSS characteristics away from the preferences of existing developers—a negative horizontal differentiation effect. FOSS are able to attract more developers when coordinators improve their level of coordination, increasing the marginal vertical network effect, or by having a more permissive Open-Source license, increasing the marginal horizontal network effect. More permissive Open-Source licenses can intensify competition between FOSS and proprietary software, resulting in lower prices. However permissive licenses may reduce the incentives to coordinate FOSS, leading to lower quality FOSS that only serve niche markets. I explore coordinators who may have different motivations—self-interested *Founders*, volunteering *Altruists*, and profit-driven *Managers*—discussing when and how they choose to coordinate FOSS.

JEL: D21, D26, L14, L17
Keywords: Open-Source Software, Network effects, Software Licensing

†Department of Economics, University of Mannheim; Mannheim Centre for Competition and Innovation.
Contact: hello@robinng.com
Acknowledgements: I am especially grateful to Paul Belleflamme, Greg Taylor, Johannes Johnen, and Rachel Tan for countless fruitful discussions. I also thank Carl-Christian Groh, Julius Goedde, Justin Johnson, Michael Kummer, François Maniquet, Jeanine Miklós-Thal, Martin Peitz, Erika Pini, Andrew Rhodes, Manpreet Singh, Yossi Spiegel, and participants of the CORE Brown Bag, Belgian Doctoral Workshop, PSE EDP Jamboree, AMES2023, MaCCI Annual Conference, IIOC2024, ZEW ICT conference, IBEO Workshop for their comments.
Funding: This project is realised with funding from: Fédération Wallonie-Bruxelles through the Action de recherche concertée grant 19/24-101; Belgian National Fund for Scientific Research through the Aspirant research fellowship FC46885; Social Science Research Council-Singapore Management University Graduate Research Fellowship; Deutsche Forschungsgemeinschaft (DFG) through the Collaborative Research Center Transregio (CRC TR) 224, Project B05.

1 Introduction

Free and Open-Source Software (FOSS) are a bedrock of the digital age. They range from prominent household names like Google's Android to niche and independently developed programs like Paint.NET. In recent years, governments have begun to recognise the importance of FOSS, providing funds to support its innovation and enhance security.[1] Despite these initiatives, and the widespread adoption of FOSS by firms and individuals, developers of FOSS rarely see monetary compensation or funding.[2] Hence, developers of FOSS are effectively volunteers, having no obligation to develop any aspect of the software. This community-driven nature of FOSS creates difficulties for those coordinating its development (Michlmayr, Hunt, & Probert, 2007).[3] Undeterred by the lack of funding and challenges, many still choose to coordinate FOSS development.

Prior research has focused on the private motives of developers contributing to a public good. However, it has overlooked the motives of coordinators leading FOSS development. By incorporating both the incentives of coordinators and developers into a theoretical model, I study how FOSS are coordinated, the features that emerge as a result of developers' preferences, and the role FOSS play in promoting competition.

Coordinators guide the development of FOSS. They act as an interface between users and developers, leading the community of developers by prioritising features and bugs (Lerner & Tirole, 2002; Klug, Bogart, & Herbsleb, 2021).[4] Higher levels of coordination provide clearer guidance to developers, ensuring that individual development efforts do not overlap, which improves the quality of the FOSS. I investigate three common motivations of coordinators (Lerner & Tirole, 2002): (i) **self-interested Founders** who are unsatisfied with existing proprietary software, seeking to build an alternative that better suits their needs, maximising their utility; (ii) **volunteering Altruists** selected to represent the views of the FOSS community, concerning themselves with the utility of members of the FOSS community (Raymond, 1999); (iii) **profit-driven Managers** providing paid features and services adjacent to the FOSS, and seeking to maximise their profits (Andersen-Gott, Ghinea, & Bygstad, 2012).[5] Considering these incentives, I explore when and how different coordinators coordinate FOSS.

[1]For example the USA's 'Securing Open Source Software Act of 2023', Europe's support for FOSS to promote Digital Sovereignty (independence from large corporations) (Paulina Grzegorzewska, 2021; Madiega, 2020; Sean Fleming, 2021).

[2]The most prominent FOSS receive funding below the USA poverty line (André Staltz, 2019; The Apache Software Foundation, 2023b). Denis Pushkarev (2023) and Marak Squires describe the funding challenges faced by developers (Emma Roth, 2022).

[3]Chris Stokel-Walker (2014); Denis Pushkarev (2023) document the problems they face in coordinating FOSS development.

[4]Raymond (1999) describes the central role of coordinators and the need for building relationships with users and developers. The Apache Software Foundation (2023a); The Document Foundation (2023); The Linux Foundation (2023) are examples of FOSS coordinated by volunteers.

[5]Prominent examples of coordinators with a profit-driven motives include: Microsoft, IBM, and Google.

Developers choose when, what, and how they contribute to FOSS. Evidence shows developers contribute to FOSS because they are attracted to the high levels of customisability FOSS provide, only contributing to features that directly benefit themselves (Raymond, 1999; Lerner & Tirole, 2002; Mustonen, 2003).[6] Hence, the resulting FOSS features depend on the preferences of developers contributing to it.

The ability of a developer to alter the features of a FOSS depends on the governance of the FOSS. I consider the software license adopted by the FOSS as a proxy for its governance structure. More restrictive licenses, such as copyleft licenses, give coordinators more control over the FOSS features.[7] This means the FOSS features are less sensitive to the preferences of developers. Conversely, permissive licenses, such as the MIT license, allow the characteristics of the FOSS to be more sensitive to developers' preferences.[8] I capture these effects in a model of horizontal product differentiation, allowing the FOSS location to depend on which developers contribute to it and the software license adopted by the FOSS.

Formally, I adopt a simple model of product differentiation à la Hotelling, where users have preferences over software features. I focus on the situation where a duopoly exists, and users choose between paying a fee and using a proprietary software or paying nothing to use the FOSS. I assume that users are also developers, and when they choose to use the FOSS, they contribute to its development.[9] Users face some mismatch cost if their preferred features differ from the features of the chosen software.

A proprietary software is located at one end of the Hotelling line and has a fixed quality. Its price is set by a profit-maximising firm. In the main model, I focus on a FOSS coordinated by it's Founder.

A Founder is located at the opposite end of the Hotelling line.[10] The Founder decides on the level of coordination of the FOSS, facing the following trade-off: On the one hand, higher levels of coordination ensure that developers' efforts complement each other, leading to distinct contributions to the FOSS, improving its quality. Through a virtuous cycle of network effects, this attracts even more developers, further improving quality.

On the other hand, the characteristics of a FOSS are endogenous and depend on who develops it. Hence, the preferences of developers affect the location of the FOSS. Because

[6] I do not differentiate between the reasons developers' preferences may differ: developers may seek solutions to a specific problem, use FOSS to signal their ability, or to build personal skills. In my model, it is sufficient that developers have some preference for the software features they work on.

[7] A FOSS with a copyleft license would require modifications and extensions to be kept as FOSS. This means proprietary software may not reuse code from copyleft software.

[8] Permissive licenses impose few, or no, restrictions on how the FOSS' source code is used or distributed.

[9] Users also serve as software testers, contributing to the task of debugging the FOSS (Raymond, 1999).

[10] Founders are motivated to develop FOSS when proprietary alternatives do not suit their needs. Hence, the most likely founder is one whose preferences differ the most from the proprietary software.

the Founder has the most extreme preferences, every additional developer shifts the FOSS away from the Founder's preferences. This increases the mismatch cost of the Founder, who may prefer lower levels of coordination to reduce his mismatch cost.

First, by considering how developers' incentives may influence the location of a FOSS, I show that FOSS generate more competitive effects than previously understood. When more developers contribute to the FOSS, the characteristics of the FOSS become closer to the proprietary software, leading to fiercer horizontal competition. This effect is in addition to previously studied scenarios where more users can improve product quality. Because more developers change the FOSS' competitive position in two dimensions, proprietary firms have a stronger incentive to lower prices and gain market share. Therefore, despite serving niche communities, the mere presence of FOSS can lead to a more competitive market.

Second, I show how the equilibrium level of coordination depends on the FOSS license. Restrictive licenses mean the FOSS location is not sensitive to the marginal developer's preference. As developers are less able to influence the final software design, they are less likely to participate in FOSS development. Fewer developers result in a lower quality FOSS. Hence, when licenses are restrictive, Founders prefer distinct contributions to boost the quality of the FOSS. Conversely, when a FOSS license is permissive, its location is more sensitive to the marginal developer, and additional developers increase the Founder's mismatch cost. As such, more permissive licenses may deter the Founder from managing FOSS, leading to a lower (or zero) level of coordination, and the underprovision of FOSS.

Third, I discuss how more permissive FOSS licenses affect the surplus in the market. More permissive FOSS licenses decrease the surplus of FOSS users. This is because permissive licenses cause the proprietary firm to set lower prices, and the Founder to lower his level of coordination. Hence, the FOSS has fewer users, and its quality decreases—both effects decreasing FOSS user surplus. Lower prices improve the surplus of proprietary software users. However, even though the firm is able to capture a larger share of the market, it receives less profit.

Finally, I repeat this analysis with volunteering Altruists and profit-driven Managers as FOSS coordinators. Altruists behave similarly to Founders. However, because they are concerned with maximising the surplus of all FOSS users, a more diverse group of users increases mismatch cost. Hence, Altruists prefer a lower level of coordination, leading to lower total surplus, than Founders. This suggests that, not withstanding other organisation skills, individual Founders are better suited to coordinate FOSS development than larger FOSS organisations.

Managers' level of coordination is binary, and depend on how competitive the proprietary software is. When the proprietary software is of a low quality, Managers prefer distinct contributions, and when the proprietary software is of a high quality, Managers prefer the lowest level of coordination that creates a viable FOSS. Comparing the three types of coordinators, I show that Managers are the only coordinator that benefit from more permissive licenses, questioning the influence that large corporations have on the Open Source Initiative, and more broadly the open-source community.

2 Related Literature

This paper studies the coordination of Free and Open-Source Software. Previous theoretical work has focused on the eccentricities of developers (Johnson, 2002, 2006; Casadesus-Masanell & Ghemawat, 2006; Athey & Ellison, 2014). My paper is most similar to Johnson (2002) which also considers how the community-driven nature of FOSS can influence the characteristics of the FOSS. Additionally, I take into account the widely reported motives of coordinators: as Founders, members of the community, or profit motivated firms (Franke & Von Hippel, 2003). Lerner and Tirole (2002) provides an overview of the literature on FOSS.

I extend the workhorse model of product differentiation provided in Hotelling (1929), which has been used to study vertical network effects, where more users can lead to higher product quality (Shy & Thisse, 1999; Fainmesser & Galeotti, 2020). In addition to vertical network effects, I introduce horizontal network effects, where users can modify the location of the product. Using this model, I study the situation where a product's compatibility with users depend on the preferences of other users. Doing so allows me to capture empirical evidence on the motives of developers who choose when, what, and how they contribute to FOSS (Fielding, 1999; Hars & Ou, 2002; Franke & Von Hippel, 2003; Shah, 2006).

There has been little theoretical work discussing the role of open-source licenses. Most papers avoid this discussion or discuss their results in the context of a specific license. Lerner and Tirole (2005b) and Gaudeul (2005) discuss simple models of license choice. Gaudeul (2008) assumes the selection of a given binary license, either permissive or restrictive, and discuss the implication of using either license. August, Chen, and Zhu (2021) argues that restrictive licenses can incentivise contributions to FOSS.

My model also takes the license choice as given, but allows for a potential continuum of license designs. I study how coordinators select the level of coordination, and hence determine the marginal network effect developers generate. In comparative statics, I show how the choice of FOSS license can affect its coordination and ability to compete with proprietary software.

In Section 3, I describe the setting in detail, discussing my main results about Founders in Section 4. I discuss the roles of Altruists and Managers in Sections 5 and 6 respectively.

In Section 7, I provide a series of extensions, and Section 8 concludes. Proofs are found in Appendix A.

3 Model

I build upon a model of product differentiation à la Hotelling (1929). My model includes 3 actors: users, a proprietary firm, and a FOSS coordinator; and 2 products: a proprietary software, and a FOSS.

User-Developers (Users). There exists a unit mass of user-developers, referred to simply as users. Users have heterogeneous preferences over the software features which are uniformly distributed along a Hotelling line, $x \sim U(0,1)$. Users choose between consuming a proprietary software or a FOSS, denoted by $i \in \{p, o\}$ respectively. The utility users receive from each software is $u_i = v_i - p_i - t|L_i - x|$, where v_i represents the software quality, p_i the price, L_i the software's location, and $t > 0$ the transport cost associated with a poor match.

For simplicity, I assume all users are developers, and model user-developers. This implies that individuals may choose to participate in the development of, and use, a FOSS, or instead simply use a proprietary software. I relax this assumption in Section 7.1, considering when users and developers are mutually exclusive groups.

Proprietary Firm. A proprietary firm produces an existing proprietary software, located at one end of the Hotelling line, $L_p = 0$. The firm's software quality is v_p, which is large enough to cover the market. It sets a price, p_p, to maximise its profits, $\pi_p = p_p \times D_p$, where D_p is demand for the firm's software.

Fixing the location and quality of the proprietary software reflects real-world development roadmaps and release cycle, and the incremental nature of software quality.[11] Moreover, this simplification allows me to focus on addressing the development of FOSS.

Free and Open-Source Software (FOSS) The quality and location of a FOSS depend on network effects. More users contributing to FOSS development leads to a quality improving vertical network effect. However, users contribute only to software features which interest them, this gives rise to a novel location shifting horizontal network effect.[12] FOSS are inherently free, $p_o = 0$.[13]

The FOSS quality depends on the number of users, $v_o = v_c + \gamma D_o$, where D_o is its demand, and v_c represents the FOSS quality if the coordinator is it's sole developer. To reduce

[11] Large software companies typically publish product roadmaps. For example, Microsoft publishes when it plans to release features, and what features are in development (Microsoft, 2023a, 2023b).

[12] Network effects arise when user utility is affected by the number of other users. My main model captures 'direct' network effects. In the general setting where users and developers are mutually exclusive, network effects are 'indirect'. See Belleflamme and Peitz (2015) for details.

[13] FOSS are free because anyone can download, build, and run its source code.

parameters, I normalise $v_c = 0$.[14] Here, $\gamma \in [0, 1]$ represents the probability that each users' contribution to the FOSS is unique. Because users are free to select what they contribute, without coordination, redundant overlapping contributions occur. Since only distinct contributions add value to the FOSS, only coordinated developments generate network effects.

The FOSS location depends on it's users' preferences, $L_o = 1 - l \times D_o$, $l \in (0, 1)$ is a parameter regulating the FOSS' location, $D_o = 1 - \bar{x}$ is the FOSS demand, where $\bar{x}$ is the marginal user indifferent between using the proprietary software and FOSS.[15] Notice that a larger l makes the location of the FOSS more sensitive to the marginal user's preferences. This reflects how more permissive FOSS licenses allow users to be more involved in the design of the FOSS.

To understand the role FOSS licenses play, consider that a permissive license guarantees the right of users to use, modify and redistribute the underlying source code. This contrasts with restrictive licenses, which may grant limited rights - limiting the ability of users to use the software.[16] The freedom permissive licenses grant users, encourages community involvement such as suggesting modifications and participating in development.[17] This way, I interpret l as the permissiveness of the license adopted by the FOSS, and suppose it is fixed ex-ante.[18]

Coordinator. FOSS development requires a coordinator to organise user contributions, higher levels of coordination ensures the distinctness of contributions, and coordinators select this probability, $\gamma \in [0, 1]$.

In the main model, I discuss a FOSS being coordinated by a self-interested **Founder** who is unsatisfied with the existing proprietary software, seeking to maximise his private utility as a user (Klug et al., 2021). Hence, the Founder is located at the opposite extreme of the Hotelling line from the proprietary software. Further, since the Founder is also a user, he maximises his own utility, $\pi_o = \gamma \times D_o - t(1 - L_o)$.[19]

The sequence of events follows: The Founder chooses the probability that contributions are distinct, γ. The proprietary firm sets its price, p_p. Users decide between the proprietary software or the FOSS. The timing of the game reflects software development lead time and the fluidity of software prices.

[14] More generally, $v_p > 0$ can be read as the difference in quality between the proprietary software and FOSS without network effects. If $v_p \leq 0$, the firm is inactive, see Etzion and Pang (2014) for details.

[15] Note $\bar{x}$ arises from the covered market assumption, discussed below.

[16] Restrictive licenses may include prohibitions to modify, use for profit, or to harm others.

[17] When licenses are permissive, other developers may 'fork' or spin-off the FOSS, this incentivises coordinators to be more responsive to community input. Denis Pushkarev (2023), the founder of core-js, a standard library for JavaScript, describes this in his regular community update.

[18] Most FOSS adopt similar licenses meeting the standard set by the Open Source Initiative (2022). GitHub (2023) also prominently promotes the MIT license. I explore this assumption in Section 7.2.

[19] This utility is identical to that of users located at 1.

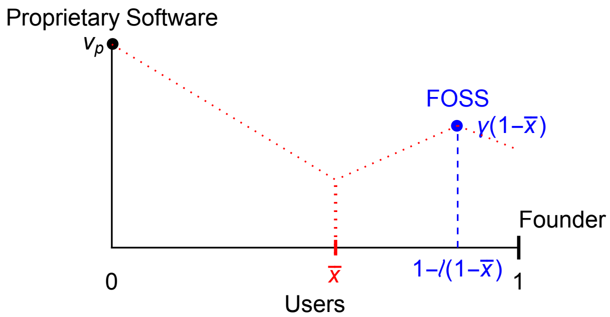


Figure 1: An illustration of the model. The endogenous location and quality of FOSS are highlighted in blue (Dashed). The utility of users is represented in red (Dotted).

Figure 1 provides a graphical representation of the model. This concludes the description of the model environment. I look for a subgame perfect Nash equilibrium under the following restrictions:

Restriction 1. *The market is covered.*

Restriction 2. *Both the firm and the FOSS are active in the market.*[20]

Since evidence suggests that many FOSS are developed in response to the needs of users which cannot be met by existing software, these restrictions allow me to focus on the situation where the proprietary software and FOSS serve as imperfect substitutes.

4 Coordinating Free and Open-Source Software

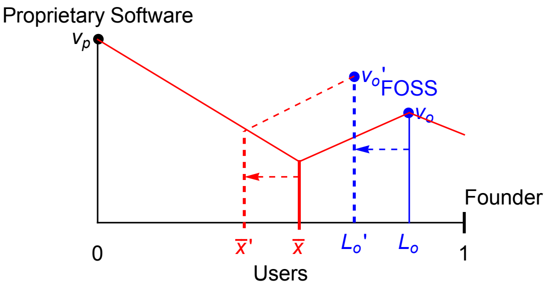
4.1 Users

In equilibrium, there exists a marginal user located at $\bar{x}$ who is indifferent between the proprietary software and the FOSS. All users to the left of $\bar{x}$ use the proprietary software, and all those to the right the FOSS. The indifferent user is pinned down by equation (1):

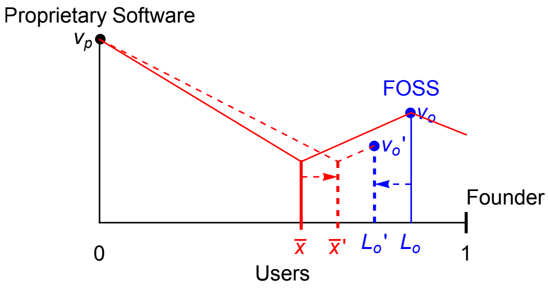
$$\begin{aligned} v_p - p_p - t\bar{x} &= \gamma D_o - p_o - t|L_o - \bar{x}| \\ v_p - p_p - t\bar{x} &= \gamma(1-\bar{x}) - p_o - t(1 - l(1-\bar{x}) - \bar{x}) \\ \bar{x} &= \frac{v_p - p_p + p_o - \gamma + t(1-l)}{t(2-l) - \gamma}. \end{aligned} \tag{1}$$

Intuitively, higher levels of coordination increases the marginal network effect ($\gamma \uparrow$), improving the quality of the FOSS, ceteris paribus, this makes the FOSS more appealing to users. To understand the role the FOSS license plays, more permissive licenses ($l \uparrow$) makes the location of the software more sensitive to the marginal user. This lowers the mismatch cost for the user when choosing to use the FOSS, ceteris paribus, permissive

[20]While not strictly necessary, this assumption allows me to study the more economically relevant situation where FOSS and proprietary software co-exist. I relax this assumption in Section 7.3.



(a) A more permissive license increases the quality of the FOSS, at fixed prices and level of coordination.



(b) Accounting for firm's pricing strategies, lower prices lead to a lower quality FOSS.

Figure 2: Dashed lines show the possible effect of a more permissive FOSS license. The FOSS is represented in blue, and user surplus in red.

licenses make a FOSS more appealing to the marginal user.

4.2 Proprietary Firm

The firm maximises its profits by setting prices. In addition to the traditional forces affecting pricing strategy in Hotelling models, network effects can cause the firm to set lower prices. These effects are summarised in Lemma 1.

Lemma 1. *The equilibrium price set by the proprietary firm strictly increases in its quality and the price of the FOSS, and strictly decreases in the level of coordination of the FOSS and permissiveness of the FOSS license.*

$$p_p^* = \frac{v_p + p_o - \gamma + t(1-l)}{2} \tag{2}$$

To understand the role of network effects, first consider how higher levels of FOSS coordination improves its quality, which leads to a virtuous cycle of network effects where a higher quality FOSS attracts even more users. This way, vertical network effects cause the firm to face a stronger quality-based competition. Second, a more permissive FOSS license means the location of the FOSS becomes more sensitive to the preferences of the marginal user. As more users join the FOSS, it's location shifts closer to the proprietary software; as such, horizontal network effects promotes location-based competition. The effect of a more permissive license, ceteris paribus, is represented in Figure 2a.

Since both network effects increase competition, the firm has a dual incentive to lower prices and prevent the FOSS userbase from becoming too large. When accounting for the price strategy of the firm, equation (1) becomes

$$\bar{x} = \frac{v_p + p_o - \gamma + t(1-l)}{2(t(2-l) - \gamma)},$$

which is increasing in both γ and l if and only if $v_p > t$. Hence, while the firm is able to reduce competitive forces through lower prices, such prices can only improve demand if the firm has a sufficiently high quality. This overall effect is illustrated in Figure 2b.

Despite being able to gain a larger market share, firms competing with highly coordinated FOSS or one with more permissive license earn less profit.

Corollary 1. *Firm's profit strictly decreases in the level of coordination of the FOSS and permissiveness of the FOSS license.*

$$\pi_p = \frac{(v_p + p_o - \gamma + t(1-l))^2}{4(t(2-l) - \gamma)}$$

These results indicate that even niche FOSS, serving small communities, can contribute to a more competitive environment. Although the firm may obtain a large market share, it lowers prices significantly to counteract the growth of the FOSS. This highlights the limitations of using market concentration as a measure of competitiveness, similar to the findings of Boone (2001).

4.3 Founder

Recall that the FOSS is free to use, $p_o = 0$, and the Founder maximise his utility:

$$\begin{aligned}\pi_o &= \gamma \times D_o - t(1 - L_o) \\ &= \frac{(\gamma - tl)(t(3-l) - v_p - \gamma)}{2(t(2-l) - \gamma)}.\end{aligned}$$

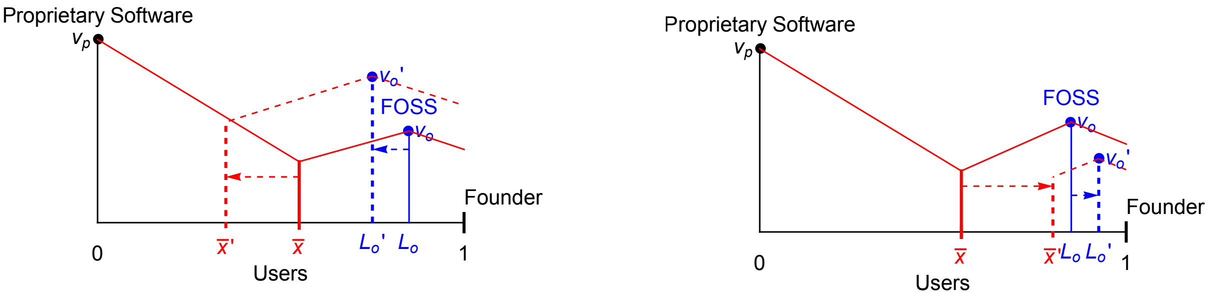
The Founder decides on the probability that contributions are distinct, γ, which proxies the level of coordination of the FOSS. This provides the first-order condition:

$$\frac{\partial \pi_o}{\partial \gamma} = \underbrace{\frac{(\gamma - tl)(t(3-l) - v_p - \gamma)}{2(t(2-l) - \gamma)^2} + \frac{t(3-l) - v_p - \gamma}{2(t(2-l) - \gamma)}}_{>0} \underbrace{- \frac{\gamma - tl}{2(t(2-l) - \gamma)}}_{<0},$$

which highlights the Founder's trade-off. On the one hand, there is an incentive to grow the network of users. Doing so leverages the contributions of additional users to improve the quality of the FOSS, and, by extension, his own utility. Hence, the Founder prefers contributions to be somewhat distinct, $\gamma > 0$. This is shown in Figure 3a.

On the other hand, however, as more users join the network, the Founder faces two opposing forces. First, the location of the FOSS shifts away from the Founder. This shift results in a software that is less compatible with the Founder's needs and preferences. Hence, the Founder has less incentive to ensure unique contributions as doing so incurs a higher mismatch cost, harming himself.

Second, the proprietary firm lowers prices in response to fiercer competition, allowing it to retain users, and minimise the benefits the FOSS receives from network effects. When



(a) At fixed prices, higher levels of coordination increase the quality of the FOSS. Although the Founder faces higher transportation cost, his surplus may increase.

(b) Accounting for firm's pricing strategies, lower prices lead to a lower quality FOSS which decreases Founder surplus.

Figure 3: Dashed lines show the possible effect of a higher level of coordination. The FOSS is represented in blue, and user surplus in red.

the quality of the proprietary firm is sufficiently high, this leads the FOSS to lose users as the Founder increases the level of coordination. Figure 3b illustrates the potential for higher levels of coordination to decrease the Founder's surplus.

Using this trade-off, I find there exists a unique subgame perfect Nash equilibrium that satisfies Restrictions 1 and 2.

Proposition 1. *Founders are active only if the proprietary software is of a sufficiently low quality, ($v_p < t(3-2l)$). Their level of coordination is weakly decreasing in the permissiveness of the FOSS license ($l \uparrow$), and the quality of the proprietary software, and strictly decreasing if contributions are not perfectly distinct ($\gamma^* < 1$).*

$$\gamma^* = min\{t(2-l) - \sqrt{2t(v_p - t)(1-l)}, 1\}$$

with $\gamma^* = 1 \iff 3t + \frac{tl^2}{2(1-l)} + \frac{1}{2t(1-l)} - \frac{2-l}{1-l} \geq v_p$ *and* $t > \frac{1}{2-l}$.[21]

Although higher levels of coordination may improve FOSS quality, proposition 1 highlights how permissive licenses can cause the Founder to limit his coordination, hindering the development of FOSS, through two channels. The first channel relates to the direct incentives faced by the Founder. Since more permissive licenses allow other developers to have a greater influence over the FOSS features, the FOSS location becomes more sensitive to the marginal user. This increases the Founder's mismatch cost, decreasing his incentive to coordinate the FOSS development. The second channel relates to market competition, where the proprietary firm lowers prices in response to a more permissive FOSS license, Lemma 1. Lower prices allows the firm to capture a larger market share, diminishing the FOSS ability to canvass network effects. The Founder, not wanting to be a victim of his own success by posing a threat to the firm, decides on a lower level of coordination.

[21] I consider $\gamma^* = 1$ to be an edge case, as it is unrealistic to have perfect coordination. Hence, my discussion in this paper focuses on the situation where $\gamma \neq 1$.

Corollary 2 contributes to a long-standing debate regarding the efficacy of permissive licenses. Proponents argue that such licenses ease the rules of participation in the FOSS community, minimising legal liability for coordinators and users, and make it easier for developers to work on FOSS (Gamblin, 2021).[22] Detractors contend that overly permissive licenses allow third parties to profiteer from public goods.[23]

Corollary 2 shows how permissive licenses may lead to an underdevelopment of FOSS, depending on market conditions (v_p and t), suggesting that differences in market conditions can explain the variety of FOSS licenses in active use. Moreover, for a given license, the activity of the Founder depends on market conditions. As market conditions evolve, Founder's activity may depend on selecting more restrictive licenses (such as in the case of Paint.NET), failing which they may halt development—prominent examples include leftpad, colors.js, faker.js, and core-js.[24] Corollary 2 connects to these anecdotes by providing a non-monetary mechanism supporting those who believe permissive FOSS licenses may lead to unsustainable software development.

This highlights the continued challenges and problems with choosing and using boilerplate, one-size-fits-all licenses— such as the MIT license (Lerner & Tirole, 2002; Subramaniam, Sen, & Nelson, 2009).[25]

Corollary 2 also shows that more permissive FOSS licenses can lead to lower quality FOSS that serve niche communities. This connects with empirical findings that many FOSS remain of a low quality, and serve only niche markets, never becoming mainstream (Lakhani & Hippel, 2004; Lerner & Tirole, 2005a; Robles, Steinmacher, Adams, & Treude, 2019). Corollary 2 explains that such niche FOSS could result from the choice of FOSS license. In particular, permissive licenses incentivises the Founder to reduce coordination, limiting the quality and scope, of the FOSS despite potential societal benefits. This result is further illustrated by anecdotal evidence that most Founders approach FOSS development with a 'pet project' mentality.

Corollary 2. *Founders only coordinate FOSS when the license is not too permissive,* $\gamma^* > 0 \iff l < \bar{l} = \frac{3t - v_p}{2t}$*. As licenses become more permissive, the quality of the FOSS decreases, and the FOSS obtains a smaller market share.*

Proof. Founders are only active if $v_p < t(3 - 2l) \iff l < \frac{3t - v_p}{2t}$.

[22] Nicolas Suzor (2013) describes on opensource.com.

[23] Ideological supporters of digital sovereignty are concerned with firms repackaging the works of FOSS (Sean Fleming, 2021; Benjamin Cedric Larsen, 2022). In a community update, Brewster (2009) Brewster explains his decision to shift Paint.NET to a more restrictive license to prevent profiteering.

[24] See Chris Williams (2016); Chris Stokel-Walker (2014) for leftpad, Emma Roth (2022) for colors.js and faker.js, and Denis Pushkarev (2023) for core-js.

[25] Almeida, Murphy, Wilson, and Hoye (2017) show how developers have a difficulty understanding the differences, and choosing, between FOSS licenses. Instead, turning to tools such as choosealicense.com and license.md. See Janis Lesinski (2020) a detailed disucssion on the problems with the MIT license.

In equilibrium, v_o and equation (1) become $v_o = \gamma(1-\bar{x})$ and $\bar{x} = \frac{v_p - \gamma^* + t(1-l)}{2(t(2-l)-\gamma^*)}$. Thus $\frac{\partial \bar{x}}{\partial l} = \frac{t(v_p-t)^2}{2(2t(v_p-t)(1-l))^{\frac{3}{2}}}$ and $\frac{\partial v_o}{\partial l} = \frac{\partial \gamma}{\partial l}(1-\bar{x}) - \frac{\partial \bar{x}}{\partial l}\gamma$, focusing on the edge case where $\gamma^* < 1$, this implies $v_p > t$ and therefore $\frac{\partial \bar{x}}{\partial l} > 0$. Since $\frac{\partial \gamma}{\partial l} \leq 0$ (Proposition 1), $\frac{\partial v_o}{\partial l} < 0$. □

Having described the trade-offs faced by the Founder, and understanding the effects FOSS licenses have on the equilibrium, I now consider how more permissive FOSS licenses can affect the distribution of surplus.

4.4 Surplus Analysis

Corollary 3. *When FOSS coordinated by Founders have more permissive licenses:*

- *User surplus generated by the FOSS strictly decreases.*
- *User surplus generated by the proprietary software strictly increases.*
- *Firm's profits strictly decreases. (Corollary 1)*

The first result in Corollary 3 describes surplus of FOSS users, and arises for two reasons. First, the decrease in surplus is directly caused by the reduction in the number of FOSS users (Corollary 2). Second, with fewer FOSS users, there is an indirect decrease in the quality of the FOSS.

A more permissive license means that more users opt for the proprietary software, leading to an increase in user surplus contributed by its consumption as l increases. Moreover, the firm lowers prices (Lemma 1), which also increases the surplus of proprietary software users. This explains the second result in Corollary 3 and highlights how the presence of a niche FOSS generates competition to the benefit of proprietary software users.

This concludes the analysis of self-interested Founders acting as coordinators of FOSS. In the following sections, I consider the behaviour of volunteering Altruist, and profit-driven Manager as coordinators of FOSS, contrasting those findings with the results about Founders.

5 Altruists

The literature notes social preferences as an important motivation for contributing to and managing FOSS: Hars and Ou (2002) suggests that developers are often motivated by building a sense of community, and Lerner and Tirole (2002) state that altruism is a main motivation reported by developers. Because many large FOSS communities are user managed, this arguably implies that such coordinators (as individuals or a group) are motivated by altruism.[26] Therefore, Alutrists are more likely to consider how their

[26] Raymond (1999) also writes about the altruistic motivation of coordinators. The Apache Software Foundation (2023a); The Document Foundation (2023); The Linux Foundation (2023) are examples of FOSS foundations governed by volunteers.

choices affects other FOSS users. To model this, I consider a coordinator that maximises the total utility of FOSS users by selecting a level of coordination.

This model follows directly from the main model, with the following modification: Instead of a self-interested Founder maximising his own utility, the coordinator is made up of volunteering Altruists seeking to maximise the value of the FOSS for its users. The FOSS continues to be free to use, $p_o = 0$. This means Altruists are maximises the total utility of FOSS users:

$$\pi_o^A = \int_{\bar{x}}^{1} u_o(x)dx = \int_{\bar{x}}^{1} v_o - t|L_o - x|dx.$$

Since this modification does not alter the behaviour of users and the proprietary firm, results about users and the proprietary firm from Sections 4.1 and 4.2 still hold. Taking into account the equilibrium decisions of users and firms,

$$\pi_o^A = \frac{(2\gamma - t(1 - 2l + 2l^2))(v_p + \gamma - t(3 - l))^2}{8(t(2 - l) - \gamma)^2}.$$

Where coordinators are Altruists, I find there exists a unique subgame perfect Nash equilibrium, satisfying Restrictions 1 and 2.

Proposition 2. *Altruists are active only if the proprietary software has a sufficiently low quality* ($v_p < t\frac{5-2l^2}{2}$)*. Their level of coordination weakly decreases in the proprietary software's quality, decreasing strictly if contributions are not perfectly distinct* ($\gamma^A < 1$)*.*

$$\gamma^A = min\left\{\frac{t(3 - 2l) + v_p - \sqrt{(v_p - t)(11t - 8tl^2 + v_p)}}{2}, 1\right\}$$

with $\gamma^A = 1 \iff v_p \in (t, \frac{1+t^2(5-3l-l^2)-t(3-2l)}{1+t(1+l-2l^2)}]$*, and* $t > \frac{1}{2-l}$.

Like the Founder, Altruists are only active if the FOSS license is sufficiently restrictive, $v_p < t\frac{5-2l^2}{2} \iff l < \sqrt{\frac{5t-2v_p}{2t}}$. Intuitively, Altruists face similar trade-offs as the Founder. However, they are concerned with minimising the total cost to users, rather than the cost of a specific user (the Founder located at 1). In particular, when more users contribute to the FOSS, total mismatch cost increases.

Unlike Founders, Altruists level of coordination is non-monotone in license permissiveness. In addition to the firm's strategy of setting lower prices in response to more permissive licenses, the adoption of a more permissive license has two contrasting effects on FOSS users. For intuition, fix the price of the proprietary software. A more permissive license causes FOSS features to be further away from 1, increasing the mismatch cost of those closer to 1. Moreover, permissive licenses attract more users, which improve the quality of the FOSS. Hence, more permissive licenses also means the marginal user has preferences for features which are more different from the FOSS. Therefore, Altruists' decision on the level of coordination is non-monotone in license permissiveness.

5.1 Surplus Analysis

Because Founders and Altruists have similar comparative statics (see Propositions 1 and 2), comparative statics on how more permissive licenses affect surplus are qualitatively identical to the situation when FOSS are coordinated by self-interested Founders. When the FOSS has a more permissive license, Corollary 4 shows that FOSS user surplus weakly decreases, proprietary software user surplus strictly increases, and the firm makes less profit.

Corollary 4. *When FOSS coordinated by Altruists have more permissive licenses:*

- *User surplus generated by the FOSS weakly decreases.*
- *User surplus generated by the proprietary software strictly increases.*
- *Firm's profits strictly decreases. (Corollary 1)*

Funding FOSS Many governments have been increasing financial support for the use and development of FOSS.[27] Comparing the surplus generated by different coordinators can provide some insight on the efficiency of how such funds are distributed.

Corollary 5 suggests that funding Founders could be more efficient than Altruists, and FOSS coordinated by Founders lead to higher total surplus. The intuition follows directly from the virtuous cycle of network effects: Higher levels of coordination improves quality, which allows the network to grow, further improving quality. Founders prefer higher levels of coordination than Altruists, thus the FOSS benefits more from these network effects. Since the surplus generated by the firm is constant, the increase in surplus generated by the FOSS causes total surplus to increase.

Corollary 5. *Founders prefer a higher level of coordination than Altruists, thus generating a higher total surplus.*

Since more permissive licenses lead a smaller demand for FOSS, Corollary 3 and 4 suggest the success of FOSS funding should not be judged by the number of FOSS users. Instead, even small and niche FOSS may create significant competitive threats, driving down prices of proprietary software; and improve user and total surplus. Additionally, Corollary 5 highlights how funding should perhaps be directly placed in the hands of Founders, instead of through umbrella funding organisations.

[27] For example to promote independence from large corporations or to ensure competition in the market (Sean Fleming, 2021; Paulina Grzegorzewska, 2021; Benjamin Cedric Larsen, 2022).

6 Profit-Driven Managers

Many prominent FOSS coordinators, such as Google and IBM, are motivated by profit. To monetise FOSS, these companies may charge for additional features and services, or earn profit through the sale of user data. In this section, I investigate how the profit-driven motive affects the coordination of FOSS.

To model profit-driven Managers as FOSS coordinators, I modify the main model to allow Managers to set prices and the level of coordination. Managers maximise $\pi_o^M = p_o \cdot D_o$, where $D_o = 1 - \bar{x}$ is the demand of the FOSS, and p_o it's price. The timing of the game follows: (i) Managers select their level of coordination, γ; (ii) Managers and the proprietary firm simultaneously set prices, p_o and p_p respectively; (iii) users decide between the FOSS or the proprietary software.

Taking into account the equilibrium decisions of users, equation (1) becomes

$$\pi_o^M = \frac{p_o(p_p + t - v_p - p_o)}{t(2-l) - \gamma}.$$

It is immediate that profits are concave in prices, and the optimal prices are

$$p_o^M = \frac{t(3-l) - \gamma - v_p}{3}$$
$$p_p^M = \frac{v_p - 2\gamma + t(3-2l)}{3}.$$

Higher levels of coordination should lead to higher quality products. Despite this, Managers choose lower prices when selecting a higher level of coordination. This is because lower prices allow Managers to attract more users, complementing the higher level of coordination. This allows FOSS quality to improve, which attracts more users, fostering a cycle of network effects. Together, both pricing strategies show how higher levels of coordination can lead to a more competitive environment, as Managers and the firm engage in fiercer price competition.

Taking into account equilibrium prices, the demand for proprietary software becomes

$$\bar{x} = \frac{t(3-2l) + v_p - 2\gamma}{3(t(2-l) - \gamma)},$$

where $\frac{\partial \bar{x}}{\partial \gamma} > 0 \iff v_p > t$. Hence, in response to higher levels of coordination by Managers, the firm is only able to use prices to gain market share if it's software has a sufficiently high quality ($v_p > t$). Lower prices allows the firm to capture a directly appeal to users, and indirectly diminish the quality, and horizontal competition, of the FOSS. However, when the quality of the proprietary software is low ($v_p < t$), the mismatch cost faced by users is too large; this diminishes the firm's ability to use prices to gain market share.

Since selecting a higher level of coordination increases the FOSS userbase if and only if the proprietary software has a sufficiently low quality, Managers only select a high level of coordination in such cases, where the gain in revenue from a larger userbase dominates any loss that arises from lower per-unit prices. Otherwise, Managers prefer lower levels of coordination, and a low marginal network effect, as this diminishes competitive forces, allowing the FOSS to have a larger userbase. Proposition 3 formalises this result.

Proposition 3. *Managers are active only if the FOSS has a sufficiently permissive license ($l \geq \frac{3t-2}{2t}$). When active, profit-driven Managers choose between binary levels of coordination:*

- *They prefer distinct contributions, $\gamma^M = 1$, when the proprietary software is of a low quality, $v_p \leq t$.*
- *They prefer the most overlap, $\gamma^M = max\{t, t(1.5 - l)\}$, when the proprietary software has a high quality, $v_p > t$.*

They prefer distinct contributions ($\gamma^M = 1$) when the proprietary software is low quality ($v_p < t$). When the proprietary software is high quality ($v_p \geq t$), Managers choose the lowest level of coordination ($\gamma^M = max\{t, t(1.5 - l)\}$) for which users continue to use the FOSS.

To open-source or not? Practically, Proposition 3 has two implications. First, when faced with costly coordination, and against a high quality proprietary software, Managers may take advantage of more permissive FOSS licenses, which allows the FOSS features to appeal to more users, while lowering their level of coordination to make more profit.

Second, and perhaps more importantly, Proposition 3 helps understand when managers may adopt an open-source business model. This can be illustrated by the competition in the internet browser market between Microsoft and Google.

Browser Wars 2.0 In 2008, the leading internet browser was Microsoft's *Internet Explorer (IE)*. This was thought to be an increasingly frustrating and outdated browser, unable to meet the growing demands by websites and users. At the same time, users were being exposed to better user interfaces and user experiences.[28] This reflects a market where $v_p < t$.

Around this time, Google adopted a code-release strategy for its new browser, *Chrome*. *Chromium*, the FOSS upon which Chrome is built, was able to benefit from the contributions of external developers, and provide a better perspective of the features users

[28]This is documented on Andriod Authority and The Verge (Luke Little, 2021; Tom Warren, 2018).

desired.[29] My model suggests that this code-release strategy can be justified by a specific market environment: where the proprietary software has a low quality, and users mismatch cost is high. In other words, $v_p < t$. When this is the case, code-releasing allows the FOSS to capture a larger market share. Therefore, capturing closely how Chrome managed to become a market leader in under 3 years.[30]

6.1 Surplus Analysis

Corollary 6 shows how profits of Managers, the firm, and users surplus change as the FOSS license becomes more permissive. Because of Managers' pricing strategies, user surplus generated by each software is non-monotone in license permissiveness, hence I am only able to describe the decrease in total user surplus.

Corollary 6. *When FOSS coordinated by Managers have more permissive licenses:*

- *Total user surplus strictly decreases.*
- *Proprietary firm's profit weakly decreases, strictly decreasing if* $l \geq 0.5$ *or* $\gamma^M = 1$. *(Corollary 1)*
- *Managers' profits:*
  - *Weakly decreases if* $\gamma^M < 1$*, strictly decreasing when* $l \geq 0.5$.
  - *Strictly increases when* $\gamma^M = 1$.

Corollary 6 shows that Managers can benefit from more permissive FOSS licenses when the proprietary software has a sufficiently low quality, and how more permissive licenses always harm users when Managers coordinate FOSS development.

Defining Open-Source Licenses Following the surplus analyses, a curious question arises: Founders and Altruists do not benefit from more permissive FOSS licenses (Corollary 3 and 4). Instead, only Managers do so. Although this is the case, The Open Source Iniative (OSI), "the authority that defines Open Source", permits only highly permissive FOSS licenses as 'Open-Source'. Hence, my model highlights the potential influence commercial institutions have on the OSI, and suggests that the OSI's incentives may be potentially skewed in favour of Managers.

7 Extensions

In this section, I provide a series of extensions, considering the situations where: (i) users and developers are mutually exclusive groups; (ii) Founders decide on a license instead of level of coordination; (iii) there exists a monopoly of either a FOSS or proprietary software; (iv) developers have heterogeneous skill levels. Proofs are in Appendix B.

[29] Making Chromium's code open-source corresponds to adopting a permissive license, $l \geq \frac{3t-2}{2t}$.
[30] As tracked by Statcounter (2022).

7.1 Non-User Developers

I begun with a model where users are also developers. Here, I consider the opposite extreme, where users and developers are mutually exclusive groups. In practice, developers are often a subset of users, and the groups do not perfectly overlap nor are they mutually exclusive. In showing these extremes, I expect that when developers are a subset of users, such a model will sit in-between the results shown here and in the main model.

In this model, users utility is $u_i = v_i - p_i - t|L_i - x|$. v_i captures the quality of the software, in particular $v_o = \gamma \cdot D_d$ where γ is the coordinator's effort to manage developers and D_d is the mass of developers.

Suppose developers exhibit preferences $o \sim U(0, 1)$ along the same Hotelling line as users. Their utility is $w_o = s_o - k|L_o - o|$, s_o captures benefit to the developer, $k > 0$ the cost associated with working on a FOSS with features they do not prefer. As suggested in the literature, developers may be motivated by skill development, and recognition both within the community and outside the community (Hars & Ou, 2002; Mustonen, 2003; Li, Tan, & Teo, 2012). The opportunities to work on complex tasks, or to gain prominence demand on the number of users of a FOSS. Hence, $s_o = \beta \cdot D_o$, where β proxies the prominence and recognition each additional FOSS user brings, and D_o the mass of users.

I focus on FOSS coordinated by a Founder, located at 1, who is motivated as a developer. Hence, his objective function is $\pi_o = \beta \cdot D_o - k(1 - L_o)$.

The timing of the game is identical to the main model, Founders choose the level of coordination, γ, proprietary firm sets the price, p_p, users and developers simultaneously decide between consuming the proprietary software or the FOSS and contributing to the FOSS respectively.

Proposition 4. *Founders only coordinate FOSS in one of the following two situations:*

- *Proprietary software is low quality, $v_p < t$ and the FOSS license is restrictive $l < \frac{1}{2}$;*
- *Proprietary software is high quality, $v_p > t$ and the FOSS license is permissive $l > \frac{1}{2}$,*

preferring distinct contributions when they do so, $\gamma^D = 1$.

Proposition 4 finds that Founders always prefer unique contributions when they are active. However, they are only active when the quality of the proprietary software is high (low) and the FOSS license more permissive (restrictive). For intuition, if the proprietary software is high quality, it is more difficult to attract users. Since vertical network effects are fixed, Founders can only use horizontal network effects to attract users, and are only active if the FOSS license is more permissive.

7.2 Founder's License Choice

In the main model, I assumed coordinators are unable to influence the license adopted by a FOSS. I consider this a realistic setting because many developers are told to adopt 'standard' FOSS licenses such as the MIT license. Moreover, unlike coordination , once a license has been selected, it is sticky. Hence, as the market environment changes, or as coordination changes hands, licenses are entrenched, and it is more interesting to focus on how the level of coordination evolves.

However, it may be interesting to understand the choice of FOSS license at its inception. To capture the initial license decision, I consider the situation where Founders does not select the level of coordination, instead selecting the permissiveness of the license adopted by the FOSS.

Proposition 5. *Founders choice of FOSS license is binary. Founders prefer the most permissive license only when the proprietary software has a sufficiently high quality and the marginal quality network effect (γ) sufficiently small.*[31] *Otherwise, Founders prefer the most restrictive license ($l = 0$).*

Proposition 5 shows that Founders only select permissive licenses when vertical network effects are too small and the FOSS relies on horizontal network effects to compete with a high quality proprietary software for users.

This result also provides some support for a recommending permissive FOSS licenses. Because FOSS sometimes face competition form high quality proprietary software, a highly permissive license might be necessary to acquire a critical mass of developers. Hence, selecting a permissive license best allows Founders to take advantage of the virtuous cycle of network effects.

7.3 Monopolistic Market

I searched for an equilibrium where a duopoly exists (Restriction 2). While this provides for a rich analysis, there exists some markets where FOSS, or proprietary software do not exist. Here, I relax Restriction 2, and describe the market conditions in which a monopoly is formed.

Corollary 7. *Founders and Altruists never form monopoly. Managers are monopolists when the quality of the proprietary software is low, $v_p \leq t(2l - 1)$, and proprietary firms are monopolists whenever the quality of the proprietary software is high, $v_p \geq t(3 - l)$.*

I show FOSS coordinated by Founders and Altruists are never able to form a monopoly. Such coordinators attempt to minimise some function of mismatch cost, and intrinsically

[31] Mathematically, $v_p > \frac{t^2(6-4l+l^2)+\gamma^2-2t\gamma(3-l)}{2(t-\gamma)}$ and $t > \gamma$.

concern themselves with the outcome of individuals with extreme preferences, located at 1. Hence, never choose to form a monopoly as this would increase the mismatch cost of those with extreme preferences.

On the other hand, Managers do not directly concern themselves with mismatch costs. Instead, they seek a balance between demand and prices, both of which decrease in v_p. Hence, when the proprietary software has a sufficiently low quality, Managers are able to capture more market share, improving the FOSS quality, and raising prices.

7.4 Skilled Users

Lerner and Tirole (2002) observes that Founders and developers of FOSS tend to be highly skilled. To reflect this observation, I propose the following modification: Developers are heterogeneous in skill level and this is uniformly distributed according to their location along the Hotelling line. In other words, developers located closer to 1 have a higher ability, normalised to $\alpha > 0$ at 1, and those further away lower ability, where the user located at 0 has 0 ability. I assume that if a developer chooses to contribute to the FOSS, their ability limits the level of their contribution.

This brings about two effects. First, the value of the software is dependent on the total skill of developers, with developers located closer to 1 generating more value, while those further away generate less value. This means the value of the FOSS is $v_o = \gamma \int_{\alpha\bar{x}}^{\alpha} \frac{x}{\alpha} dx = \gamma \frac{\alpha(1-\bar{x}^2)}{2}$. Second, because developers with a higher skill level are more likely to contribute to more components/critical components of the software, the location of the software is more affected by their output. This means the FOSS is located at $L_o = 1 - l \int_{\alpha\bar{x}}^{\alpha} \frac{x}{\alpha} dx = 1 - \frac{\alpha(1-\bar{x}^2)}{2}$.

Proposition 6. *When users have heterogeneous skill levels, Founders prefer unique contributions, $\gamma^S = 1$.*

Proposition 6 suggests that when there are heterogeneous skill levels, with more skilled developers located further away from the proprietary software, Founders prefers unique contributions. This is because the location of the software shifts less for each additional developer than in the base model. As such, the mismatch cost to Founders increases at a lower rate than the base model, and Founders prefer more distinct contributions.

8 Conclusion

This paper provides a first look at a model which accounts for the diverse motivations of participants in Free and Open-Source Software development. I show that FOSS Founders and Altruistic members of the FOSS community only coordinate FOSS if the FOSS license is sufficiently restrictive. When they do so, Founders prefer higher levels of coordination, suggesting they are more effective at coordinating FOSS than Altruists. Given limited resources, FOSS funders should carefully consider the intrinsic motivations of coordinators, and may choose to prioritise Founders over Altruists.

Profit-driven Managers, however, only coordinate FOSS if its license is sufficiently permissive. Managers prefer high levels of coordination if the FOSS competes with a low quality proprietary software, but only a minimal level of coordination when the proprietary software is high quality. This finding aligns closely with the development of Chrome and Android, potentially explaining when large corporations choose to participate in and how they coordinate FOSS development.

Most FOSS adopt permissive licenses that conform to the standards prescribed by the Open Source Initiative. This paper shows that only Managers benefit from more permissive FOSS licenses, while coordinators with other incentives and FOSS users can be harmed. Hence, suggesting that the promotion and extensive use of one-size-fits-all, highly permissive licenses, might reflect corporate influence on the FOSS community. Indeed, many corporations are heavily invested into FOSS which can be seen in the widespread effect of the Log4j vulnerability (Michael Hill, 2022), the xkcd comic on the dependence on FOSS[32], and Microsoft's development of the internet browser Edge[33] Additionally, the normalisation of highly permissive licenses leads to the underdevelopment of FOSS by Founders and Altruists; to foster their contributions to FOSS, the use of customised licenses, tailored to each FOSS, should be encouraged instead.

My model considers how developing FOSS features affects it's location, but does not account for how software features can be complementary, or how coordinators may employ other mechanisms to limit the FOSS location. Further, a static model allows for many tractable results, but ignores the dynamic nature of FOSS development where developers can fork (spin-off) an existing FOSS into a competing software. I also abstract from an environment with costly coordination which would reduce the level of FOSS coordination. To better understand these and other nuances related to FOSS development, different approaches may need to be taken.

At its heart, through the use of horizontal network effects, this model captures how collective outcomes are shaped by individual inputs. In addition to FOSS, vertical and horizontal network effects co-exist in many other applications such as the development of large language models, social movements and political activism, crowd-sourcing and resource sharing, and online communities. I leave such applications for future research.

[32]See the comic here https://xkcd.com/2347/.

[33]Microsoft adopted, and became a main contributor of, Chromium in the lead up to the launch of Microsoft Edge, marking a new era of Browser wars.

References

Almeida, D. A., Murphy, G. C., Wilson, G., & Hoye, M. (2017). Do software developers understand open source licenses? In *2017 ieee/acm 25th international conference on program comprehension (icpc)* (pp. 1–11).

Andersen-Gott, M., Ghinea, G., & Bygstad, B. (2012). Why do commercial companies contribute to open source software? *International journal of information management*, *32*(2), 106–117.

André Staltz. (2019). *Software below the poverty line.* Retrieved 2023-05, from `https://staltz.com/software-below-the-poverty-line.html`

Athey, S., & Ellison, G. (2014). Dynamics of open source movements. *Journal of Economics & Management Strategy*, *23*(2), 294–316.

August, T., Chen, W., & Zhu, K. (2021). Competition among proprietary and open-source software firms: The role of licensing in strategic contribution. *Management Science*, *67*(5), 3041–3066.

Belleflamme, P., & Peitz, M. (2015). *Industrial organization: markets and strategies.* Cambridge University Press.

Benjamin Cedric Larsen. (2022). *The geopolitics of ai and the rise of digital sovereignty.* Retrieved 2023-03, from `https://www.brookings.edu/articles/the-geopolitics-of-ai-and-the-rise-of-digital-sovereignty/`

Boone, J. (2001). Intensity of competition and the incentive to innovate. *International Journal of Industrial Organization*, *19*(5), 705–726.

Brewster, R. (2009). *A new license for paint.net v3.5.* Retrieved 2023-03, from `https://blog.getpaint.net/2009/11/06/a-new-license-for-paintnet-v35/`

Casadesus-Masanell, R., & Ghemawat, P. (2006). Dynamic mixed duopoly: A model motivated by linux vs. windows. *Management Science*, *52*(7), 1072–1084.

Chris Stokel-Walker. (2014). *The internet is being protected by two guys named steve.* Retrieved 2023-05, from `https://www.buzzfeed.com/chrisstokelwalker/the-internet-is-being-protected-by-two-guys-named-st`

Chris Williams. (2016). *How one developer just broke node, babel and thousands of projects in 11 lines of javascript.* Retrieved 2023-05, from `https://www.theregister.com/2016/03/23/npm_left_pad_chaos/`

Denis Pushkarev. (2023). *2023-02-14-so-whats-next.md.* Retrieved 2023-05, from `https://github.com/zloirock/core-js/blob/master/docs/2023-02-14-so-whats-next.md`

Emma Roth. (2022). Open source developer corrupts widely-used libraries, affecting tons of projects. *The Verge*. Retrieved 2022-10, from `https://www.theverge.com/2022/1/9/22874949/developer-corrupts-open-source-libraries-projects-affected`

Etzion, H., & Pang, M.-S. (2014). Complementary online services in competitive markets: Maintaining profitability in the presence of network effects. *Mis Quarterly*, *38*(1), 231–248.

Fainmesser, I. P., & Galeotti, A. (2020). Pricing network effects: Competition. *American Economic Journal: Microeconomics*, *12*(3), 1–32.

Fielding, R. T. (1999). Shared leadership in the apache project. *Communications of the ACM*, *42*(4), 42–43.

Franke, N., & Von Hippel, E. (2003). Satisfying heterogeneous user needs via innovation toolkits: the case of apache security software. *Research policy*, *32*(7), 1199–1215.

Gamblin, T. (2021). *Picking an open source license at llnl: Guidance and recommendations from the computing directorate* (Tech. Rep.). Livermore, CA (United States): Lawrence Livermore National Lab.(LLNL).

Gaudeul, A. (2005). Public provision of a private good: What is the point of the bsd license? *Available at SSRN 933631*.

Gaudeul, A. (2008). Open source licensing in mixed markets, or why open source software does not succeed. *CCP Working Paper*.

GitHub. (2023). *Choose an open source license.* Retrieved 2023-05, from `https://choosealicense.com/`

Hars, A., & Ou, S. (2002). Working for free? motivations for participating in open-source projects. *International journal of electronic commerce*, *6*(3), 25–39.

Hotelling, H. (1929). Stability in competition. *The Economic Journal*, *39*(153), 41–57.

Janis Lesinski. (2020). *The mit licence is community hostile.* Retrieved 2023-05, from `https://www.lesinskis.com/MIT-licence-community-hostile.html`

Johnson, J. P. (2002). Open source software: Private provision of a public good. *Journal of Economics & Management Strategy*, *11*(4), 637–662.

Johnson, J. P. (2006). Collaboration, peer review and open source software. *Information Economics and Policy*, *18*(4), 477–497.

Klug, D., Bogart, C., & Herbsleb, J. D. (2021). "they can only ever guide" how an open source software community uses roadmaps to coordinate effort. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW1), 1–28.

Lakhani, K. R., & Hippel, E. v. (2004). How open source software works: "free" user-to-user assistance. In *Produktentwicklung mit virtuellen communities* (pp. 303–339). Springer.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The journal of industrial economics*, *50*(2), 197–234.

Lerner, J., & Tirole, J. (2005a). The economics of technology sharing: Open source and beyond. *Journal of Economic Perspectives*, *19*(2), 99–120.

Lerner, J., & Tirole, J. (2005b). The scope of open source licensing. *Journal of Law, Economics, and Organization*, *21*(1), 20–56.

Li, Y., Tan, C.-H., & Teo, H.-H. (2012). Leadership characteristics and developers' motivation in open source software development. *Information & Management*, *49*(5), 257–267.

Luke Little. (2021). *Google chrome: Its history and rise to market domination.* Retrieved 2022-11, from https://www.androidauthority.com/google-chrome-history-1025602/

Madiega, T. A. (2020). Digital sovereignty for europe. *EPRS: European Parliamentary Research Service*.

Michael Hill. (2022). *The apache log4j vulnerabilities: A timeline.* Retrieved 2023-03, from https://www.csoonline.com/article/571797/the-apache-log4j-vulnerabilities-a-timeline.html

Michlmayr, M., Hunt, F., & Probert, D. (2007). Release management in free software projects: Practices and problems. In *Open source development, adoption and innovation: Ifip working group 2.13 on open source software, june 11–14, 2007, limerick, ireland 3* (pp. 295–300).

Microsoft. (2023a). *Microsoft 365 roadmap.* Retrieved 2023-05, from https://www.microsoft.com/en-gb/microsoft-365/roadmap?filters=

Microsoft. (2023b). *Microsoft release cycle.* Retrieved 2023-05, from https://learn.microsoft.com/en-us/windows/deployment/update/release-cycle

Mustonen, M. (2003). Copyleft—the economics of linux and other open source software. *Information Economics and Policy*, *15*(1), 99–121.

Nicolas Suzor. (2013). *What motivates free software developers to choose between copyleft and permissive licences?* Retrieved 2023-05, from https://opensource.com/law/13/8/motivation-free-software-licensing

Open Source Initiative. (2022). *Licenses & standards.* Retrieved 2022-11, from https://opensource.org/licenses

Paulina Grzegorzewska. (2021). *Parlimentary mission supports open source.* Retrieved 2023-03, from https://joinup.ec.europa.eu/collection/open-source-observatory-osor/news/french-report-digital-sovereignty

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23–49.

Robles, G., Steinmacher, I., Adams, P., & Treude, C. (2019). Twenty years of open source software: From skepticism to mainstream. *IEEE Software*, *36*(6), 12–15.

Sean Fleming. (2021). *What is digital sovereignty and why is europe so interested in it?* Retrieved 2023-03, from https://www.weforum.org/agenda/2021/03/europe-digital-sovereignty/

Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management science*, *52*(7), 1000–1014.

Shy, O., & Thisse, J.-F. (1999). A strategic approach to software protection. *Journal of*

Economics & Management Strategy, *8*(2), 163–190.

Statcounter. (2022). *Desktop browser market share worldwide.* Retrieved 2022-11, from https://gs.statcounter.com/browser-market-share/desktop/worldwide#monthly-200901-202210

Subramaniam, C., Sen, R., & Nelson, M. L. (2009). Determinants of open source software project success: A longitudinal study. *Decision Support Systems*, *46*(2), 576–585.

The Apache Software Foundation. (2023a). *Apache corporate governance.* Retrieved 2023-03, from https://www.apache.org/foundation/governance/

The Apache Software Foundation. (2023b). *Foundation reports and statements.* Retrieved 2023-05, from https://www.apache.org/foundation/reports.html

The Document Foundation. (2023). *The document foundation.* Retrieved 2023-03, from https://www.documentfoundation.org/governance/bodies/

The Linux Foundation. (2023). *Linux foundation leadership teams.* Retrieved 2023-03, from https://www.linuxfoundation.org/about/leadership

Tom Warren. (2018). *Google's chrome browser is now 10 years old.* Retrieved 2022-11, from https://www.theverge.com/2018/9/2/17811844/google-chrome-browser-10-years-history

Appendix A Proofs

Proof of Lemma 1. To find the equilibrium price of the proprietary software,

$$\begin{aligned}\pi_p &= p_p \cdot \bar{x} \\ &= p_p \frac{v_p - p_p + p_o - \gamma + t(1-l)}{t(2-l) - \gamma}\end{aligned}$$

Notice that for an interior profit maximising solution, $t(2-l) - \gamma > 0$ is required. This ensures that $\frac{\partial^2 \pi_p}{\partial p_p^2} < 0$, and implies that $v_p - p_p + p_o - \gamma + t(1-l) > 0$.

Therefore, $p_p^* = \frac{v_p + p_o - \gamma + t(1-l)}{2}$. Restriction 2 then implies

$$t(2-l) - \gamma > 0 \tag{3}$$

$$v_p + p_o + t(1-l) - \gamma > 0 \tag{4}$$

$$t(3-l) - p_o - v_p - \gamma > 0 \tag{5}$$

I now show that p_p^* increases in v_p and p_o, and decreases in γ and l:

$$\frac{\partial p_p^*}{\partial v_p} = \frac{1}{2} > 0, \qquad \frac{\partial p_p^*}{\partial p_o} = \frac{1}{2} > 0, \qquad \frac{\partial p_p^*}{\partial \gamma} = -\frac{1}{2} < 0, \qquad \frac{\partial p_p^*}{\partial l} = -\frac{t}{2} < 0.$$

This concludes the proof. □

Proof of Corollary 1. To show Corollary 1, recall $\pi_p = p_p \times \bar{x}$. Substituting $\bar{x}$ and p_p^*,

$$\begin{aligned}\pi_p &= \frac{[v_p + p_o - \gamma + t(1-l)]^2}{4(t(2-l) - \gamma)} \\ \frac{\partial \pi_p}{\partial \gamma} &= \frac{v_p + p_o + t(1-l) - \gamma}{4(t(2-l) - \gamma)^2}(v_p + p_o + \gamma - t(3-l)) \\ \frac{\partial \pi_p}{\partial l} &= t\frac{v_p + p_o + t(1-l) - \gamma}{4(t(2-l) - \gamma)^2}(v_p + p_o + \gamma - t(3-l)) = t\frac{\partial \pi_p}{\partial \gamma}\end{aligned}$$

First notice from the restrictions imply $t(2-l) - \gamma > 0$ and $v_p + p_o + t(1-l) - \gamma > 0$. This means that the sign of $\frac{\partial \pi_p}{\partial \gamma}$ depends on the sign of $v_p + p_o + \gamma - t(3-l)$. From equation (5), $v_p + p_o + \gamma - t(3-l) < 0$. Therefore $\frac{\partial \pi_p}{\partial \gamma} < 0$, by extension $\frac{\partial \pi_p}{\partial l} < 0$.

This concludes the proof. □

Proof of Proposition 1. I solve the game by backward induction. Notice that in the Proof of Lemma 1, I have solved the last two stages of the game. To proof Proposition 1, I look at the first stage of the game and propose a candidate equilibrium, ruling out all other possible equilibrium. I then show that such an equilibrium satisfies the restrictions, and maximises Founder's utility.

Recall that $\pi_o = (\gamma - tl)(1-\bar{x})$. Considering equation (1), and remembering that $p_o = 0$,

$$\pi_o = \frac{(\gamma - tl)(t(3-l) - v_p - \gamma)}{2(t(2-l) - \gamma)}$$

Taking the first and second order conditions,

$$\begin{aligned}
\frac{\partial \pi_o}{\partial \gamma} &= \frac{t(t(6-6l+l^2) - 2v_p(1-l)) + \gamma^2 - 2\gamma t(2-l)}{2(t(2-l)-\gamma)^2} \\
&\text{evaluated at } \gamma = 0, \\
&= \frac{t(6-6l+l^2) - 2v_p(1-l)}{2t(2-l)^2}, > 0 \text{ at } t(3 + \frac{l^2}{2(1-l)}) > v_p \\
\frac{\partial^2 \pi_o}{\partial \gamma^2} &= -\frac{2t(1-l)(v_p - t)}{(t(2-l)-\gamma)^3}, < 0 \text{ at } t(2-l) - \gamma > 0.
\end{aligned}$$

Notice that $\gamma \in [0,1]$ implies the FOSS is only active if $3t + \frac{tl^2}{2(1-l)} > v_p$.

Solving for γ^*

$$\gamma^* = t(2-l) \pm \sqrt{2t(v_p - t)(1-l)}$$

Notice that $\sqrt{2t(v_p-t)(1-l)}$ is real only if $v_p \geq t$, because $l \in (0,1)$. And when $\sqrt{2t(v_p-t)(1-l)}$ is real, it is ≥ 0. From Restriction 2, $t(2-l) > \gamma$, therefore I reject $t(2-l) + \sqrt{2t(v_p-t)(1-l)}$ as a candidate for γ^*. Moreover, because $\gamma \in [0,1]$, $\gamma^* = 1$ when $t > \frac{1}{2-l}$ and $v_p \leq \frac{1-2t(2-l)+t^2(6-6l+l^2)}{2t(1-l)} = 3t + \frac{tl^2}{2(1-l)} + \frac{1}{2t(1-l)} - \frac{2-l}{1-l}$. In all other cases, $\gamma^* = t(2-l) - \sqrt{2t(v_p-t)(1-l)}$. Therefore, I propose $\gamma^* = min\{t(2-l) - \sqrt{2t(v_p-t)(1-l)}, 1\}$ as a candidate solution.

I show when this solution satisfies Restriction 1, Consider that the Founder and the indifferent user must have weakly positive utility. Founder's utility is given by π_o, and this is weakly positive whenever $\gamma \geq tl$. To see that $\gamma^* \geq tl$,

$$\begin{aligned}
t(2-l) - \sqrt{2t(v_p-t)(1-l)} &\geq tl \\
2t(1-l) &\geq (v_p - t) \\
\frac{3t - v_p}{2t} &\geq l
\end{aligned}$$

and $1 \geq tl \iff l \leq \frac{1}{t}$ when $\gamma^* = 1$.

The indifferent user's utility is $\frac{(\gamma - t(1-l))(t(3-l)-v_p-\gamma)}{2(t(2-l)-\gamma)}$. Because Restriction 2 imply $t(2-l) - \gamma > 0$ and $t(3-l) - v_p - \gamma > 0$, it remains that the indifferent user's utility is weakly positive if $\gamma \geq t(1-l)$. To see that $\gamma^* \geq t(1-l)$,

$$t(2-l) - \sqrt{2t(v_p - t)(1-l)} \geq t(1-l)$$
$$t \geq 2(v_p - t)(1-l)$$
$$l \geq \frac{2v_p - 3t}{2(v_p - t)}$$

and $1 \geq t(1-l) \iff l \geq 1 - \frac{1}{t}$ when $\gamma^* = 1$.

Combining $\frac{3t - v_p}{2t} \geq l$ with $l \geq \frac{2v_p - 3t}{2(v_p - t)}$,

$$\frac{3t - v_p}{2t} \geq \frac{2v_p - 3t}{2(v_p - t)}$$
$$v_p(2t - v_p) \geq 0$$

requires $2t \geq v_p$.

Together, this means that when $\gamma^* = t(2-l) - \sqrt{2t(v_p - t)(1-l)}$, $l \in [\frac{2v_p - 3t}{2(v_p - t)}, \frac{3t - v_p}{2t}]$ such that $2t \geq v_p$. And $1 - \frac{1}{t} \leq l \leq \frac{1}{t}$, which is true when $t \in [1, 2]$.

Next, I show when this solution satisfies Restriction 2. It is immediate that the interior candidate solution always satisfies equation (3). To see it satisfies equation (4),

$$v_p + t(1-l) - t(2-l) + \sqrt{2t(v_p - t)(1-l)} > 0$$
$$v_p - t + \sqrt{2t(v_p - t)(1-l)} > 0$$

This is always true.

To see it satisfies (5),

$$t(3-l) - v_p - t(2-l) + \sqrt{2t(v_p - t)(1-l)} > 0$$
$$t - v_p + \sqrt{2t(v_p - t)(1-l)} > 0$$
$$2t(1-l) > v_p - t$$
$$t(3 - 2l) > v_p$$

And the corner solution of $\gamma^* = 1$ satisfies these equations when $t(2-l) > 1$, $v_p + t(1-l) > 1$, and $t(3-l) - v_p > 1$.

I now show that the candidate solution is utility maxinimising.

At $\gamma = 0$, $\frac{\partial \pi_o}{\partial \gamma}$ is $\frac{t(6 - 6l + l^2) - 2v_p(1-l)}{2t(2-l)^2}$. This is positive whenever $t(3 + \frac{l^2}{2(1-l)}) > v_p$. Recall from equation (5) that $t + \sqrt{2t(v_p - t)(1-l)} > v_p$. I now show directly that $t(3 + \frac{l^2}{2(1-l)}) \geq$

$t + \sqrt{2t(v_p - t)(1-l)}$:

$$t(3 + \frac{l^2}{2(1-l)}) \geq t + \sqrt{2t(v_p - t)(1-l)}$$
$$t(2 + \frac{l^2}{2(1-l)}) \geq \sqrt{2t(v_p - t)(1-l)}$$

Recall $t(3-2l) > v_p \iff 2t(1-l) > v_p - t$. This means that $2t(1-l) > \sqrt{2t(v_p - t)(1-l)}$ and showing $t(2 + \frac{l^2}{2(1-l)}) \geq 2t(1-l)$ implies that $t(2 + \frac{l^2}{2(1-l)}) > \sqrt{2t(v_p - t)(1-l)}$.

$$t(2 + \frac{l^2}{2(1-l)}) \geq 2t(1-l)$$
$$\frac{4(1-l) + l^2}{2(1-l)} \geq 2(1-l)$$
$$4 \geq 3l$$

Since $l \in (0,1)$, this is always true, and $\frac{\partial \pi_o}{\partial \gamma} > 0$ at $\gamma = 0$.

It is immediate that $\frac{\partial^2 \pi_o}{\partial \gamma^2} < 0$ for the range of possible solutions. This is because it is negative when $v_p > t$. And equation (3) requires this.

Therefore, the candidate solution maximises Founder's utility.

I now show the Founder does not always play $\gamma = 1$ when it is active. For the Founder to play $\gamma^* = 1$, it must be that $3t + \frac{tl^2}{2(1-l)} + \frac{1}{2t(1-l)} - \frac{2-l}{1-l} \geq 3t + \frac{tl^2}{2(1-l)}$, and $t > \frac{1}{2-l}$. Suppose to a contradiction that the Founder always plays $\gamma^* = 1$.

The first criteria reduces to

$$3t + \frac{tl^2}{2(1-l)} + \frac{1}{2t(1-l)} - \frac{2-l}{1-l} \geq 3t + \frac{tl^2}{2(1-l)}$$
$$\frac{1 - 2t(2-l)}{t(1-l)} \geq 0$$
$$\frac{1}{2(2-l)} \geq t$$

and the second implies

$$t > \frac{1}{2-l}$$
$$\frac{1}{2(2-l)} > \frac{1}{2-l}$$
$$\frac{1}{2} > 1.$$

A contradiction. This means that there exists a range of v_p and t for which Founders will play the interior solution $t(2-l)-\sqrt{2t(v_p-t)(1-l)}$.

Therefore, I conclude that there exists a unique equilibrium where $\gamma^* = min\{t(2-l) - \sqrt{2t(v_p-t)(1-l)}, 1\}$ maximises Founders' utility subject to Restrictions 1 and 2, which require $l \in [\frac{2v_p-3t}{2(v_p-t)}, \frac{3t-v_p}{2t}]$. In particular, $\gamma^* = 1$ if $v_p \leq 3t + \frac{tl^2}{2(1-l)} + \frac{1}{2t(1-l)} - \frac{2-l}{1-l}$ and $t > \frac{1}{2-l}$, and $l \in [1-\frac{1}{t}, \frac{1}{t}]$.

To see that $\frac{\partial\gamma^*}{\partial l} \leq 0$ and $\frac{\partial\gamma^*}{\partial v_p} \leq 0$, at $\gamma^* = t(2-l) - \sqrt{2t(v_p-t)(1-l)}$,

$$\frac{\partial\gamma^*}{\partial l} = \frac{t(v_p - t - \sqrt{2t(v_p-t)(1-l)})}{\sqrt{2t(v_p-t)(1-l)}}$$
$$\frac{\partial\gamma^*}{\partial v_p} = -\frac{t(1-l)}{\sqrt{2t(v_p-t)(1-l)}}$$

From equation (5), $t - v_p + \sqrt{2t(v_p-t)(1-l)} > 0$, hence $\frac{\partial\gamma^*}{\partial l} < 0$. And $l \in (0,1)$ implies $\frac{\partial\gamma^*}{\partial v_p} < 0$.

When $\gamma^* = 1$, changes to l or v_p does not affect γ^*. Therefore $\frac{\partial\gamma^*}{\partial l} = 0$ and $\frac{\partial\gamma^*}{\partial v_p} = 0$.

This concludes the proof. □

Before embarking on the surplus analysis, I provide the expressions for consumer surplus, CS_i where $i \in \{o, p, b\}$ represent the FOSS, the proprietary software and the total consumer surplus respectively.

$$\begin{aligned} CS_o &= \int_{\bar{x}}^{1} \gamma(1-\bar{x}) - t|(1-l(1-\bar{x}) - x| - p_o dx \\ &= \frac{(1-\bar{x})^2(2\gamma + 2tl(1-l) - t)}{2} - p_o(1-\bar{x}) \qquad (6) \\ CS_p &= \frac{\bar{x}(2v_p - t\bar{x} - 2p_p)}{2} \qquad (7) \\ CS_b &= CS_o + CS_p \qquad (8) \end{aligned}$$

Proof of Corollary 3. Evaluating CS_o and CS_p in the Founders equilibrium, recall that

$$\begin{aligned} p_o &= 0 \\ \bar{x} &= \frac{v_p - \gamma^* + t(1-l)}{2(t(2-l)-\gamma^*)} \\ \gamma^* &= t(2-l) - \sqrt{2t(v_p-t)(1-l)} \\ p_p^* &= \frac{v_p + p_o - \gamma^* + t(1-l)}{2}. \end{aligned}$$

Therefore, from equations (6) and (7), $\frac{\partial CS_o}{\partial l} < 0$ and $\frac{\partial CS_p}{\partial l} > 0$ respectively.

This concludes the proof. □

Proof of Proposition 2. I solve the game by backward induction, and the last two stages are identical to that of Proposition 1. Hence, I rely on those results to proof the first stage of the game.

To show the first stage, I first propose a candidate equilibrium, ruling out all other possible equlibria. I then show that the candidate equilibrium satisfies the Restrictions, and maximises Altruists' utility.

Recall that $\pi_o^A = \int_{\bar{x}}^1 \gamma(1-\bar{x}) - t|1-l(1-\bar{x})-x|dx$. Accounting for equation (1), $\pi_o^A = \frac{(2\gamma - t(1-2l+2l^2))(v_p+\gamma-t(3-l))^2}{8(t(2-l)-\gamma)^2}$.

Taking the first order condition,

$$\frac{\partial \pi_o^A}{\partial \gamma} = \frac{(v_p+\gamma-t(3-l))(\gamma(v_p+t(3-2l)-\gamma)+t(v_p(1+l-2l^2)-t(5-3l-l^2)))}{4(t(2-l)-\gamma)^3}$$
$$\frac{\partial^2 \pi_o^A}{\partial \gamma^2} = \frac{(v_p-t)(4t\gamma(1-l^2)-t^2(17-4l-14l^2+4l^3)+2\gamma v_p+tv_p(5+2l-6l^2))}{4(t(2-l)-\gamma)^4}$$

First, I find the candidate γ^A. Solving $\frac{\partial \pi_o^A}{\partial \gamma}=0$,

$$\gamma^A = \begin{cases} t(3-l)-v_p \\ \frac{t(3-2l)+v_p-\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2} \\ \frac{t(3-2l)+v_p+\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2} \end{cases}$$

I eliminate the first candidate because it violates equation (5). I eliminate the third candidate because it leads to a contradiction, consider $v_p+t(1-l)-\gamma>0$ then,

$$\begin{aligned} v_p+t(1-l)-\frac{t(3-2l)+v_p+\sqrt{(v_p-t)(11t-8tl^2+v_p}}{2} &> 0 \\ \frac{v_p-t-\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2} &> 0 \\ (v_p-t)^2 > (v_p-t)(11t-8tl^2+v_p) \\ 8tl^2 &> 12t \\ 2l^2 &> 3 \end{aligned}$$

This is not possible because $l \in (0,1)$. This leaves $\frac{t(3-2l)+v_p-\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2}$ as the only candidate. Since $(11t-8tl^2+v_p)>0$, a real solution only exists if $v_p>t$.

I now show that at the candidate $\gamma^A = \frac{t(3-2l)+v_p-\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2}$ that $\frac{\partial \pi_o^A}{\partial \gamma}<0$. Notice that $\frac{\partial \pi_o^A}{\partial \gamma}>0$ only if $v_p < t\frac{5-2l^2}{2}$ and $\gamma > \frac{t(t(17-4l-14l^2+4l^3)-v_p(5+2l-6l^2))}{2(v_p+2(1-l^2)t)}$, and $\frac{\partial \pi_o^A}{\partial \gamma}<0$

whenever $v_p > t\frac{5-2l^2}{2}$ or $\gamma < \frac{t(t(17-4l-14l^2+4l^3)-v_p(5+2l-6l^2))}{2(v_p+2(1-l^2)t)}$.

I now show that the candidate $\gamma^A < \frac{t(t(17-4l-14l^2+4l^3)-v_p(5+2l-6l^2))}{2(v_p+2(1-l^2)t)}$ when $v_p < t\frac{5-2l^2}{2}$:

$$\begin{aligned}
\frac{(v_p-t)(11t-8tl^2+v_p)}{v_p+2t(1-l^2)} &< \sqrt{(v_p-t)(11t-8tl^2+v_p)} \\
\frac{(v_p-t)(11t-8tl^2+v_p)}{(v_p+2t(1-l^2))^2} &< 1 \\
(v_p-t)(11t-8tl^2+v_p) &< (v_p+2t(1-l^2))^2 \\
t(3-2l^2)(2v_p+t(-5+2l^2)) &< 0 \text{ notice that } 3-2l^2>0 \\
2v_p-t(5-2l^2) &< 0
\end{aligned}$$

Therefore, whenever $v_p < t\frac{5-2l^2}{2}$, $\gamma^A < \frac{t(t(17-4l-14l^2+4l^3)-v_p(5+2l-6l^2))}{2(v_p+2(1-l^2)t)}$. And I conclude that there is no scenario where $\frac{\partial^2\pi_o^A}{\partial\gamma^2} > 0$ subject to my restrictions. The candidate equilibrium maximises Altruists' utility.

Notice that there exists some range where $\gamma^A = 1$. This upper bound is reached when $\frac{t(3-2l)+v_p-\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2} \geq 1 \iff v_p \leq \frac{1+t^2(5-3l-l^2)-t(3-2l)}{1+t(1+l-2l^2)}$ and $t > \frac{1}{2-l}$.

I conclude that Altruists always selects a positive level of coordination, $\gamma^A > 0$. And $\gamma^A = min\{\frac{t(3-2l)+v_p-\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2}, 1\}$ and Altruists are active when $v_p < t\frac{5-2l^2}{2}$.

To see that $\frac{\partial\gamma^A}{\partial v_p} \leq 0$, at $\frac{t(3-2l)+v_p-\sqrt{(v_p-t)(11t-8tl^2+v_p)}}{2}$,

$$\begin{aligned}
\frac{\partial\gamma^A}{\partial v_p} &= \frac{1}{2} - \frac{v_p+t(5-4l^2)}{2\sqrt{(v_p-t)(11t-8tl^2+v_p)}} \\
\frac{1}{2} - \frac{v_p+t(5-4l^2)}{2\sqrt{(v_p-t)(11t-8tl^2+v_p)}} &< 0 \\
0 &< t^2(3-2l^2)^2
\end{aligned}$$

this is always true because $t > 0$ and $l \in (0,1)$.

And when $\gamma^A = 1$, changes to v_p does not affect γ^A. Therefore, $\frac{\partial\gamma^A}{\partial v_p} \leq 0$.

This concludes the proof. □

Proof of Corollary 4. Evaluating CS_o and CS_p in the Altruists equilibrium, recall that

$$\begin{aligned} p_o &= 0 \\ \bar{x} &= \frac{v_p - \gamma^A + t(1-l)}{2(t(2-l)-\gamma^*)} \\ \gamma^A &= \frac{t(3-2l) + v_p - \sqrt{(v_p - t)(11t - 8tl^2 + v_p)}}{2} \\ p_p^* &= \frac{v_p + p_o - \gamma^A + t(1-l)}{2} \end{aligned}$$

Therefore, from equations (6) and (7), $\frac{\partial CS_o}{\partial l} \leq 0$ and $\frac{\partial CS_p}{\partial l} > 0$ respectively.

This concludes the proof. □

Proof of Corollary 5. To show Corollary 5, I compare the total surplus in the market at γ^A and γ^*.

The total surplus in the market is given as

$$\begin{aligned} TS &= CS_b + p_p \cdot \bar{x} \\ &= \frac{(1-\bar{x})^2(2\gamma - t(1-2l+2l^2)) + \bar{x}(2v_p - t\bar{x})}{2} \end{aligned}$$

Recall $t(2-l) - \gamma > 0$, $v + t(1-l) - \gamma > 0$, $t(3-l) - v - \gamma > 0$. Therefore, evaluating TS at γ^* and γ^A shows $TS(\gamma^*) > TS(\gamma^A)$.

This concludes the proof. □

To show Proposition 3, I first show Lemma 2 which describes the interaction between Managers and firm's pricing strategies.

Lemma 2. *The pricing strategy of profit-driven Managers is* $p_o^M = \frac{t(3-l)-\gamma-v_p}{3}$ *and proprietary firm is* $p_p^M = \frac{v_p - 2\gamma + t(3-2l)}{3}$.

Proof of Lemma 2. To show Lemma 2, recall that both Managers and the proprietary firm set prices simultaneously, and firm's pricing strategy is given in Lemma 1. I solve for Managers' pricing strategy, then show the equilibrium pricing strategy.

Managers' profit function is given by $\pi_o^M = p_o \cdot (1-\bar{x})$, where $\bar{x}$ is given by equation (1),

$$\pi_o^M = p_o \frac{t + p_p - p_o - v_p}{t(2-l) - \gamma}$$

and profits are concave in price. Therefore,

$$p_o^M = \frac{p_p + t - v_p}{2}$$

accounting for $p_p = \frac{v_p + p_o - \gamma + t(1-l)}{2}$,

$$p_o^M = \frac{t(3-l) - \gamma - v_p}{3}$$
$$p_p^M = \frac{v_p - 2\gamma + t(3-2l)}{3}$$

This concludes the proof. □

Further, accounting for the pricing strategies, Restriction 2 implies that

$$t(2-l) - \gamma > 0 \tag{9}$$
$$v_p + t(3-2l) - 2\gamma > 0 \tag{10}$$
$$t(3-l) - v_p - \gamma > 0 \tag{11}$$

Using these pricing strategies, I now show Proposition 3.

Proof of Proposition 3. I solve the game by backward induction, where the last two stages are given by Lemma 1 and 2. To solve the first stage, I first propose a candidate equilibrium which maximises Managers' profits. I then show how such an equilibrium satisfies the model restrictions.

Accounting for the pricing strategies, the indifferent user is $\bar{x} = \frac{v_p + t(3-2l) - 2\gamma}{3(t(2-l)-\gamma)}$ and Managers' profit is now

$$\pi_o^M = \frac{(t(3-l) - v_p - \gamma)^2}{9(t(2-l) - \gamma}$$
$$\frac{\partial \pi_o^M}{\partial \gamma} = \frac{(t(3-l - v_p - \gamma)(\gamma - v_p - t(1-l))}{9(t(2-l) - \gamma)^2}$$
$$\frac{\partial^2 \pi_o^M}{\partial \gamma^2} = \frac{2(v_p - t)^2}{9(t(2-l) - \gamma)^3} \text{ , } > 0$$

First, observe that within the restrictions of the model, equations (9) and (11) imply that it is always profitable for Managers to coordinate the FOSS.

Second, observe that the second order condition implies that the FOSS profit is not concave in the level of coordination. Therefore, whenever $\frac{\partial \pi_o^M}{\partial \gamma} > 0$, Managers prefer the highest level of coordination. Since $t(3-l) - \gamma - v_p > 0$ and $t(2-l) - \gamma > 0$, $\frac{\partial \pi_o^M}{\partial \gamma} > 0 \iff \gamma - v_p - t(1-l) > 0$ and 0 with equality.

Suppose first that $v_p < t$. I show that $\frac{\partial \pi_o^M}{\partial \gamma} > 0$. From $\gamma - v_p - t(1-l) > 0$,

$$\begin{aligned} \gamma - v_p - t(1-l) &> 0 \text{ adding } t(2-l) - \gamma > 0, \\ t(2-l) - \gamma + \gamma - v_p - t(1-l) &> 0 \\ t &> v_p \end{aligned}$$

Therefore, whenever $v_p < t$, $\frac{\partial \pi_o^M}{\partial \gamma} > 0$ and Managers prefer the highest level of coordination, $\gamma^M = 1$.

Suppose next that $v_p \geq t$. When this is true, I show that $\frac{\partial \pi_o^M}{\partial \gamma} < 0$. Suppose to a contradiction that $\gamma - v_p - t(1-l) > 0$.

$$\begin{aligned} \gamma - v_p - t(1-l) &> 0 \\ t - v_p &> t(2-l) - \gamma \end{aligned}$$

the left hand side is weakly negative, and the right hand side is positive. Hence, a contradiction. This implies that $\frac{\partial \pi_o^M}{\partial \gamma} < 0$, and Managers are still active, but prefer the lowest level of coordination.

I now consider the restrictions of the model. Recall that for Restriction 1 the indifferent user and the extreme user located at 1 must have weakly positive utility, and for Restriction 2 the equations (9), (10) and (11) hold.

The indifferent user's utility is given by

$$u_o(\bar{x}) = (1-\bar{x})(\gamma - t(1-l)) - p_o = \frac{(t(3-l) - v_p - \gamma)(2\gamma - t(3-2l))}{3(t(2-l) - \gamma)},$$

and the for the extreme user $u_o(1) = \frac{2(t(3-l)-v_p-\gamma)(\gamma-t)}{3(t(2-l)-\gamma))}$. For the indifferent user to receive weakly positive utility, $\gamma \geq \frac{t(3-2l)}{2}$, and for the extreme user, $\gamma \geq t$.

Because $\gamma \in [0,1]$, Restriction 1 implies that $1 \geq \frac{t(3-2l)}{2}$ or $l \geq \frac{3t-2}{2t}$ is a sufficient condition for Managers to be active and the market to be covered.

I now show the restrictions are satisfied by the candidate equilibrium.

Suppose first that $v_p < t$, and the candidate equilibrium is $\gamma^M = 1$. The following equations need to be satisfied.

$$\begin{aligned}1 &\geq t\\ 1 &\geq \frac{t(3-2l)}{2}\\ t(2-l) &> 1\\ v_p + t(3-2l) &> 2\\ t(3-l) - v_p &> 1\end{aligned}$$

Observe that when $v_p < t$ then whenever $t(2-l) > 1$ is satisfied, both of $v_p + t(3-2l) > 2$ and $t(3-l) - v_p > 1$ are satisfied. Notice that $1 \geq t$ is consistent whenever

$$\begin{aligned}&t(2-l) > 1 \geq t\\ &t(2-l) > t\\ &t(1-l) > 0\end{aligned}$$

which is always true because $t > 0$ and $l \in (0, 1)$.

Finally, $1 \geq \frac{t(3-2l)}{2} \iff 2 \geq t(3-2l)$. And this is consistent whenever

$$\begin{aligned}v_p + t(3-2l) &> 2 \geq t(3-2l)\\ v_p + t(3-2l) &> t(3-2l)\\ v_p &> 0\end{aligned}$$

And this is implicitly true, or firms are inactive. See Etzion and Pang (2014) for details.

Therefore, whenever Restriction 1 and 2 are satisfied and $v_p < t$, Managers choose $\gamma^M = 1$.

Suppose next that $v_p \geq t$, and the candidate equilibrium is the smallest level of γ that satisfies the following equations.

$$\begin{aligned}\gamma &\geq t\\ \gamma &\geq \frac{t(3-2l)}{2}\\ t(2-l) &> \gamma\\ v_p + t(3-2l) &> 2\gamma\\ t(3-l) - v_p &> \gamma\end{aligned}$$

Notice first that $\gamma \geq t$ and $\gamma \geq \frac{t(3-2l)}{2}$ imply that for the candidate equilibrium to satisfy Restriction 1, $\gamma^M = max\{t, t(1.5-l)\}$. Observe that $\gamma^M = t$ if $l \geq 0.5$ and $\gamma^M = t(1.5-l)$ if $l < 0.5$.

Using this, I show that the candidate equilibrium satisfies Restriction 2.

Suppose first that $l \geq 0.5$ then (9), (10) and (11) become

$$t(2-l) > t$$
$$v_p + t(3-2l) > 2t$$
$$t(3-l) - v_p > t$$

which are all satisfied because

$$t(1-l) > 0$$
$$v_p + t(1-l) > 0$$
$$t(2-l) - v_p > 0$$

are always true because $t > 0$ and $l \in [0.5, 1)$.

Suppose next that $l < 0.5$ then (9), (10) and (11) become

$$t(2-l) > t(1.5-l)$$
$$v_p + t(3-2l) > t(3-2l)$$
$$t(3-l) - v_p > t(1.5-l)$$

which are all satisfied because

$$0.5t > 0$$
$$v_p > 0$$
$$1.5t - v_p > 0$$

which are always true because $t > 0$.

Therefore, following Restrictions 1 and 2, $l \geq \frac{3t-2}{2t}$ is a sufficient condition for Managers to be active. And when $v_p \geq t$, $\gamma^M = \max\{t, t(1.5-l)\}$, when $v_p < t$, $\gamma^M = 1$.

This concludes the proof. □

Proof of Corollary 6. I first show that total consumer surplus (CS_b) decreases in l, then show how the profits of both proprietary firm and profit-driven Managers decreases in l.

To see this, I evaluate each term in equilibrium, taking the first derivative with respect to l.

Recall that

$$p_o^M = \frac{t(3-l) - \gamma^M - v_p}{3}$$
$$p_p^M = \frac{v_p - 2\gamma^M + t(3-2l)}{3}$$
$$\bar{x} = \frac{v_p + t(3-2l) - 2\gamma^M}{3(t(2-l) - \gamma^M)}$$
$$\gamma^M = \begin{cases} 1 & \text{if } v_p < t \\ max\{t, t(1.5-l)\} & \text{if } v_p \geq t \end{cases}$$

I show that the total consumer surplus in unambiguously decreasing in the permissiveness of the FOSS license.

$$CS_b = CS_o + CS_p$$
$$\frac{\partial CS_b}{\partial l} \begin{cases} < 0 & \text{when } \gamma^M = 1 \\ = \frac{2(t^2(-5+9l-6l^2+l^3)+tv_p(3-2l+l^2)-v_p^2)}{9t(1-l)^2} < 0 & \text{when } \gamma^M = t \text{ and } l \in [0.5, 1) \\ = -\frac{2l(3t-2v_p)^2}{9t} < 0 & \text{when } \gamma^M = t(1.5-l) \end{cases}$$

I now show that proprietary firm profit decreases in the permissiveness of the FOSS license.

$$\pi_p^M = \frac{(v_p + t(3-2l) - 2\gamma)^2}{9(t(2-l) - \gamma)}$$
$$\frac{\partial \pi_p^M}{\partial l} = \begin{cases} \frac{t(t(3-2l)-v_p-2)(2+v_p-t(5-2l)))}{9(t(2-l)-1)^2} < 0 & \text{if } \gamma^M = 1 \\ -\frac{(t(1-2l)+v_p)(t(3-2l)-v_p)}{9t(1-l)^2} < 0 & \text{if } \gamma^M = t \\ 0 & \text{if } \gamma^M = t(1.5-l) \end{cases}$$

Finally I show how Managers profits change in the permissiveness of the FOSS license, and that this depends on the their choice of γ.

$$\pi_o^M = \frac{(t(3-l) - v_p - \gamma)^2}{9(t(2-l) - \gamma)}$$
$$\frac{\partial \pi_o^M}{\partial l} = \begin{cases} \frac{(t(3-l)-v_p-1)(1-v_p-t(1-l))}{9(t(2-l)-1)^2} > 0 & \text{if } \gamma^M = 1 \\ \frac{(tl-v_p)(t(2-l)-v_p)}{9t(1-l)^2} < 0 & \text{if } \gamma^M = t \\ 0 & \text{if } \gamma^M = t(1.5-l) \end{cases}$$

This concludes the proof. □

Appendix B Proofs (Extensions)

Proof of Proposition 4. I use backward induction to solve for the equilibrium. This is done in the following steps: (1) determine the marginal user indifferent between consumption of the proprietary software and the FOSS; (2) determine the marginal developer indifferent between other leisure and the development of the FOSS; (3) solve for the price strategy of the firm; (4) solve for the optimal level of coordination for developers of the FOSS (γ^D).

In deciding which software to use, users compare between the following utilities, $u_p = v_p - p_p - tx$ and $u_o = \gamma(1-\bar{o}) - t|(1 - l(1-\bar{o}) - x)|$. The demand here arises from the uniform distribution. The indifferent user being $\bar{x}$, and $\bar{o}$ the indifferent developer.

$$\begin{aligned} v_p - p_p - t\bar{x} &= (1-\bar{o})(\gamma + tl) - t(1-\bar{x}) \\ \bar{x} &= \frac{v_p - p_p - (1-\bar{o})(\gamma + tl) + t}{2t} \end{aligned}$$

From Restriction 1, $\bar{x}$ gives the demand for the proprietary software, and $(1-\bar{x})$ the demand of the FOSS. I turn now to the decision of the developers.

Developers decide between contributing to the FOSS and receiving a utility of $w_o = s_o - k|L_o - o| = \beta(1-\bar{x}) - k|(1 - l(1-\bar{o}) - o)|$. Thus the indifferent developer is

$$\begin{aligned} \beta(1-\bar{x}) - k(1 - l(1-\bar{o}) - \bar{o}) &= 0 \\ \bar{o} &= \frac{\beta(\bar{x}-1) + k(1-l)}{k(1-l)} \\ &= 1 - \frac{\beta(1-\bar{x})}{k(1-l)} \end{aligned}$$

This means that

$$\begin{aligned} \bar{x} &= \frac{k(1-l)(v+t-p) - \beta(\gamma + tl)}{2tk(1-l) - \beta(\gamma + tl)} \\ \bar{o} &= \frac{2tk(1-l) + \beta(v - p - t(1-l) - \gamma)}{2tk(1-l) - \beta(\gamma + tl)} \end{aligned}$$

We will need $k > \frac{\beta(\gamma+tl)}{2t(1-l)}$ for concavity of firm's profit function.

Now we turn to the firm's decision.

$$\pi_p = p_p\bar{x}$$
$$p_p = \frac{1}{2}(v_p + t - \frac{\beta(\gamma + tl)}{k(1-l)})$$
$$\bar{x} = \frac{k(1-l)(v+t) - \beta(\gamma+tl)}{2(2tk(1-l) - \beta(\gamma+tl))}$$
$$\bar{o} = \frac{4tk^2(1-l)^2 + \beta^2(\gamma+tl) + k\beta(1-l)(v_p + 2tl - 3t - 2\gamma)}{2k(1-l)(2tk(1-l) - \beta(\gamma+tl))}$$
$$\pi_p = \frac{((v+t)k(1-l) - \beta(\gamma+tl))^2}{4k(1-l)(2tk(1-l) - \beta(\gamma+tl))}$$

Finally, we turn our attention to the coordinator of the FOSS. Recall that Founders are selfish, and is only motivated by maximising its individual utility.

$$\pi_o = \beta(1-\bar{x}) - kl(1-\bar{o})$$
$$= \frac{\beta(\beta(2l-1)(\gamma+tl) + k(1-l)(t(3-6l+4l^2) + v_p(2l-1)))}{2(1-l)(2tk(1-l) - \beta(\gamma+lt))}$$
$$\frac{\partial\pi_o}{\partial\gamma} = \frac{\beta^2k(4tl^2 + (2l-1)(v_p - t))}{2(2tk(1-l) - \beta(\gamma+tl))^2)}$$
$$\frac{\partial^2\pi_o}{\partial\gamma^2} = \frac{\beta^3k(4tl^2 + (2l-1)(v_p - t))}{(2tk(1-l) - \beta(\gamma+tl))^3}$$

Observe that both $\frac{\partial\pi_o}{\partial\gamma}$ and $\frac{\partial^2\pi_o}{\partial\gamma^2} > 0 \iff$ both $2l-1>0$ and $v_p - t > 0$ simultaneously. Hence, the FOSs is only active if there is a sufficiently permissive license when the proprietary software has a high quality, and a sufficiently restrictive license when the proprietary software has a low quality. In either case, Founders select $\gamma^D = 1$ preferring that contributions by each developer is unique.

Notice that β does not influence the outcome of this decision.

This concludes the proof.

□

Proof of Proposition 5. To show this, I solve the game by backward induction. Since the last two stages of the game is identical to the main model, the proof for the indifferent user and proprietary firm's action can be found in Lemma 1. Recall that the objective function of Founders is

$$\pi_o = \gamma(1-\bar{x}) - t(l(1-\bar{x}))$$
$$= (\gamma - tl)\left(\frac{t(3-l)-\gamma-v_p}{2(t(2-l)-\gamma)}\right)$$
$$\frac{\partial \pi_o}{\partial l} = \frac{-t(\gamma^2 + t((6-4l+l^2)t-2v) + 2\gamma((-3+l)t+v_p))}{2(t(2-l)-\gamma)^2}$$
$$\frac{\partial^2 \pi_o}{\partial l^2} = \frac{2t^2(v_p-t)(t-\gamma)}{(t(2-l)-\gamma)^3}.$$

If $\gamma \geq t$, then $\frac{\partial \pi_o}{\partial l} \leq 0$.

If $\gamma < t$, then $\frac{\partial^2 \pi_o}{\partial l^2} > 0$. And $\frac{\partial \pi_o}{\partial l} > 0 \iff v_p > \frac{t^2(6-4l+l^2)+\gamma^2-2t\gamma(3-l)}{2(t-\gamma)}$.

This concludes the proof. □

Proof of Corollary 7. I first show that Founders and Altruists never form monopolies. I then show the conditions which allow profit-driven Managers to form a monopoly, and a proprietary firm to form a monopoly.

To see that Founders and Altruists never form monopolies, observe that $\bar{x} = \frac{v_p-\gamma+t(1-l)}{2(t(2-l)-\gamma)}$. For $\bar{x} \leq 0$, $v_p + t(1-l) - \gamma \leq 0$. This means $\gamma \geq v_p + t(1-l)$ is necessary for the FOSS to form a monopolist. At the interior solutions for both Founders and Altruists, this is not true.

To see when Managers form a monopoly, observe that $\bar{x} = \frac{v_p+t(1-2l)}{t(4-3l)-\gamma}$. For $\bar{x} \leq 0 \iff v_p \leq t(2l-1)$.

To see when a proprietary firm is a monopolist, observe that $\bar{x} \geq 1 \iff v_p \geq t(3-l)-\gamma$. By definition of monopoly, $\gamma = 0$. Therefore a firm forms monopoly whenever $v_p \geq t(3-l)$.

This concludes the proof.

□

Proof of Proposition 6. This game is identical to the main model, with the following change: users have heterogeneous skill level, and those located at 1 have a higher skill level than those located at 0. I solve this game by backward induction to find γ^S, the optimal unique contributions for Founders.

First, I look at the decision made by the marginal user.

$$v_p - p_p - t\bar{x} = \frac{(1-\bar{x})(\alpha(1+\bar{x})(\gamma+tl)-2t)}{2}$$

From this, we can find partial derivatives for $\bar{x}$ with respect to p_p and γ.

$$\frac{\partial \bar{x}}{\partial p_p} = \frac{1}{\bar{x}\alpha(\gamma + tl) - 2t}$$
$$\frac{\partial \bar{x}}{\partial \gamma} = \frac{\alpha(1 - \bar{x}^2)}{2(\bar{x}\alpha(\gamma + tl) - 2t)}$$

Solving explicitly for $\bar{x}$,

$$\bar{x} = \frac{2t \pm \sqrt{4t^2 + \alpha^2(\gamma + tl)^2 - 2\alpha(\gamma + tl)(v + t - p)}}{\alpha(\gamma + tl)}$$

Moving to the next stage, we look at the decision made by the firm. Notice that for any real interior solution, this means that $\frac{\partial \bar{x}}{\partial p} < 0$. This implies that $2t > \bar{x}\alpha(\gamma + tl)$. And additionally that we reject the positive of $\bar{x}$, and $\bar{x} = \frac{2t - \sqrt{4t^2 + \alpha^2(\gamma+tl)^2 - 2\alpha(\gamma+tl)(v+t-p)}}{\alpha(\gamma+tl)}$. This means for positive demand, $p < v + t - \frac{\alpha(\gamma+tl)}{2}$ and for a real solution to $\bar{x}$, $p \geq v + t - \frac{4t^2+\alpha^2(\gamma+tl)^2}{4\alpha(\gamma+tl)}$. Hence, for a duopoly it must be that $2t > \alpha(\gamma + tl)$.

Solving for the optimal price,

$$\pi_p = p\bar{x}$$
$$\frac{\partial \pi_p}{\partial p} = \bar{x} + \frac{\partial \bar{x}}{\partial p} p$$

For a solution, $\frac{\partial \pi_p}{\partial p} = 0$ and this implies that $p_p = -\frac{\bar{x}}{\frac{\partial \bar{x}}{\partial p_p}} = \bar{x}(2t - \bar{x}\alpha(\gamma + tl))$.

$$\frac{\partial^2 \pi_p}{\partial p_p^2} = 2\frac{\partial \bar{x}}{\partial p} + p\frac{\partial^2 \bar{x}}{\partial p^2}$$
$$= \frac{3\bar{x}\alpha(\gamma + tl) - 4t}{(2t - \bar{x}\alpha(\gamma + tl))^2} < 0$$

For an interior solution, $3\bar{x}\alpha(\gamma + tl) - 4t < 0$ or $\bar{x} < \frac{4t}{3\alpha(\gamma+tl)}$ or $\gamma < t(\frac{4}{3\bar{x}\alpha} - l)$.

These two conditions together imply that $2t > \alpha(\gamma + tl)$ or that $\alpha < \frac{2t}{\gamma+tl}$.

Together, the three conditions $\alpha < \frac{4t}{3\bar{x}(\gamma+tl)}$, $\alpha < \frac{2t}{\bar{x}(\gamma+tl)}$ and $\alpha < \frac{2t}{\gamma+tl}$ suggest that an interior solution is only possible if α is sufficiently small, bound above by $\frac{2t}{\gamma+tl}$ if $\bar{x} \leq \frac{2}{3}$ and $\frac{4t}{3\bar{x}(\gamma+tl)}$ if $\bar{x} > \frac{2}{3}$. In all other cases, there will be zero demand for the proprietary software. That is, if users are exceptionally skilled, they simply create a unique product on their own, driving out the proprietary firm. Given this, I now turn to the Founders' choice. Observe that $\alpha < \frac{2t}{\bar{x}(\gamma+tl)}$ is the least restrictive condition. Also notice that rewriting $\bar{x} < \frac{4t}{3\alpha(\gamma+tl)}$ and $\bar{x} < \frac{2t}{\alpha(\gamma+tl)}$ suggests that $\bar{x}$ decreases in α.

Moving to the first stage, the decision by Founders is

$$\pi_o = \frac{\alpha(\gamma - tl)(1 - \bar{x}^2)}{2}$$

And subject to $\gamma \leq 1$, letting λ be the shadow price,

$$\begin{aligned}\frac{\partial \pi_o}{\partial \gamma} &= \frac{\alpha(1 - \bar{x}^2 - 2(\gamma - tl)\bar{x}\frac{\partial \bar{x}}{\partial \gamma})}{2} - \lambda \\ &= \frac{\alpha t(1 - \bar{x}\alpha l)(1 - \bar{x}^2)}{2t - \bar{x}\alpha(\gamma + tl)} - \lambda\end{aligned}$$

$\lambda \geq 0$, $\lambda(1 - \gamma) = 0$, $1 - \gamma \geq 0$.

Notice that for $\frac{\alpha t(1-\bar{x}\alpha l)(1-\bar{x}^2)}{2t-\bar{x}\alpha(\gamma+tl)} - \lambda = 0$, either $\bar{x} = \frac{1}{\alpha l}$ and $\lambda = 0$ or $\lambda = \frac{\alpha t(1-\bar{x}\alpha l)(1-\bar{x}^2)}{2t-\bar{x}\alpha(\gamma+tl)}$. In the first case, $\lambda = 0$ implies that this solution can be true for any γ. In the second case, $\lambda > 0$ implies that $\gamma = 1$.

Observe that the coordinator is only active if $\gamma > tl$. Suppose that $\gamma \neq 1$. Then $\frac{\partial \pi_o}{\partial \gamma} \geq 0 \iff (1 - \bar{x}\alpha l) \geq 0$, and 0 only with equality This means should there be an interior solution for the FOSS where $\bar{x} = \frac{1}{\alpha l}$.

However, I show that $\bar{x} = \frac{1}{\alpha l}$ leads to a contradiction between an active FOSS, $\gamma > tl$, and an active firm, $2t - \bar{x}\alpha(\gamma + tl) = 2t - \frac{\alpha(\gamma+tl)}{\alpha l} > 0 \iff tl > \gamma$. Hence, a duopoly cannot exist if $\bar{x} = \frac{1}{\alpha l}$.

Therefore, the optimal solution for Founders is $\gamma^S = 1$.

I conclude that skilled self-interested Founders choose to develop a FOSS if $1 > tl$, and they prefer that contributions of each developer is unique, $\gamma^S = 1$.

This concludes the proof. □