Quality Regulation on Two-Sided Platforms: Exclusion, Subsidization, and First-Party Applications

Peng Huang[1], Gaoyan Lyu[2], and Yi Xu[1]

[1]*Robert H. Smith School of Business, University of Maryland, College Park*

[2]*School of Management and Economics, Beijing Institute of Technology, Beijing*

January 4, 2022

Abstract

Managing the quality of complementary applications is vital to the success of a two-sided platform. While prior research has focused solely on restricting platform access based on a quality threshold, we compare three quality regulation strategies: (1) the platform excludes access to low-quality complementors, (2) it provides a fixed amount of subsidy to high-quality complementors, and (3) it develops its own high-quality applications in addition to those from third-party complementors. Our analyses reveal that the widely adopted exclusion strategy is a special case of the subsidization strategy, and it does not always benefit the platform. In contrast, both subsidization and first-party applications strategies render the platform owner better off, with higher profits, higher average quality, and a larger consumer network, but only subsidization always improves social welfare. In addition, the trade-off between subsidization and first-party applications strategies depends on the development cost of first-party applications and the fraction of high-quality complementors, but the relationship is not monotonic. Our results demonstrate that the platform does not have to sacrifice application quantity for higher application quality. With the right choices, it can profitably improve both measures simultaneously. This research provides concrete guidelines to help platform managers make decisions about regulating the quality of complementary applications.

1 Introduction

Firms in technology industries often build their product or service offerings around a platform, consisting of a set of core elements that are used in common across implementations along with interchangeable, complementary components that enhance the value of the platform (Boudreau, 2010). This mechanism of value co-creation gives rise to the model of platform ecosystems, where the success of a platform depends critically on coordinating third-party complementary innovations (Ceccagnoli et al., 2012; Gawer & Cusumano, 2002). However, to orchestrate such a platform ecosystem, firms face significant governance challenges, such as balancing platform openness and control (Boudreau, 2010), providing boundary resources (Ghazawneh & Henfridsson, 2013), and managing intellectual properties within the ecosystem (Huang et al., 2013; Parker & Van Alstyne, 2017). A burgeoning body of literature has examined a variety of issues involved in the governance of technology platforms, particularly in the context of those serving two or more distinct user groups in the presence of network effects (Eisenmann et al., 2009; Gawer & Henderson, 2007; Hagiu, 2014; Parker & Van Alstyne, 2005; Song et al., 2018; Tiwana et al., 2010).

Despite progress, an understudied but fundamentally important issue in platform governance is regulation of the quality of complementary applications (Hagiu, 2009a). The significance of quality regulation is highlighted by the collapse of the video game market in the early 1980s, where unrestricted entry resulted in a market for "lemons." Poor-quality games flooded the market, leading to the bankruptcy of over 90% of video game developers and the failure of the dominant video game platform at the time, Atari (Boudreau & Hagiu, 2009). In contrast, the later success of Nintendo was partly attributed to its restrictive platform access strategy. Nintendo used a security chip to lock out unlicensed, low-quality game developers. Recent technology platforms have witnessed a number of more subtle quality regulation strategies. Although denying access to low-quality complementors is still widely adopted (e.g., Apple's iOS platform), some have embraced a strategy that subsidizes high-quality complementors. For example, to attract high-quality complementors, Google offered $10 million in prizes to developers of the best apps in early stages of its Android platform,[1] and Facebook created the fbFund in partnership with venture capitalists, which awarded seed grants to selected start-ups dedicated to developing Facebook applications.[2] In addition, many two-sided platform owners, such as manufacturers of video game consoles (e.g., PlayStation, Xbox) and media streaming service providers (e.g., Netflix, Hulu, Amazon Prime Video), often create their own high-quality applications or content—also known as first-party applications—on top of their platforms (Hagiu and Spulber 2013).[3] These exclusive applications, sometimes offered as part of a product bundle, play an important role in attracting an initial critical mass of platform adopters and thus winning the battle with competing platforms, especially when third-party applications are subject to multihoming (Hagiu & Spulber, 2013; Rochet & Tirole, 2003).

Although some scholars have started to tackle the issue of quality regulation on two-sided platforms with network effects, extant research in this area has focused primarily on the strategy of exclusion based on a quality threshold (Hagiu, 2009a; Zheng & Kaiser, 2013). Given the varied quality regulation strategies employed by recent platforms, there is a notable gap in understanding the relative effectiveness and limitations of these strategies. We aim to address this gap by analyzing a model under the setting of a profit-maximizing two-sided platform, where consumer utility depends not only on the variety of complementary applications[4] but also on their quality. In our model, applications developed by

[1] See http://googlepress.blogspot.com/2007/11/google-announces-10-million-android_12.html.

[2] See https://techcrunch.com/2007/09/17/facebook-launches-fbfund-with-accel-and-founders-fund-to-invest-in-new-facebook-apps/.

[3] Some examples of first-party applications/content include the *Halo* franchise by Xbox, the *Uncharted* franchise by PlayStation, the *House of Cards* series by Netflix, *The Handmaid's Tale* series by Hulu, and the movie *Manchester by the Sea* by Amazon Prime Video.

[4] In this work, we use the terms "application variety" and "application quantity" interchangeably.

complementors differ from one another both vertically and horizontally, and their indirect network effect parameter is a function of application quality. The platform owner collects revenue by charging entry fees to both sides of the market. We compare three quality regulation mechanisms: (1) the platform excludes low-quality complementors using a quality threshold, (2) it provides a fixed subsidy to high-quality complementors, and (3) it produces high-quality first-party applications/content at a cost, thus improving the average quality of applications in the platform ecosystem.

Our analyses yield several important observations. First, we show that the widely adopted exclusion strategy is a special case of the subsidization strategy; that is, for every optimal exclusion strategy, there is an equivalent subsidization strategy that achieves the same level of profit. However, there are conditions under which exclusion is strictly dominated by the subsidization strategy, which is more flexible due to its mechanism of price discrimination. Second, compared with the benchmark scenario in which there is no platform owner intervention, both subsidization and first-party applications strategies render the platform owner better off, with higher profits, higher average quality, a larger consumer network, and a higher consumer access fee, but only subsidization always improves social welfare. Importantly, in contrast to the exclusion strategy (Hagiu, 2009a), the adoption of the other two strategies does not require sacrificing quantity to improve quality (or vice versa); indeed, both strategies can achieve a greater quantity *and* a higher quality of applications at the same time. Third, the trade-off between subsidization and first-party applications strategies depends on the development cost of first-party applications and the fraction of high-quality complementors, but the relationship is not monotonic. Comparing the two, the winning strategy is always associated with a larger consumer network, but not necessarily a higher average quality. Finally, we discuss the limitation of each quality regulation strategy. For subsidization, the disadvantage becomes more apparent when the fraction of high-quality complementors is particularly low or high, which leads to cost inefficiency and limited effectiveness in improving quality. For first-party applications, the platform faces difficulty internalizing the development cost, primarily due to the free-riding of low-quality complementors. In response, the platform may choose to exclude outside participation altogether if the market is fraught with low-quality complementors or if it faces a sufficiently low development cost, resulting in a vertically integrated platform.

We further examine the robustness of the findings by relaxing some assumptions of the model, such as allowing third-party application development costs to be dependent on application quality or using a concave first-party application development cost function instead of a convex one. While most of the results continue to hold, we gain some additional insights.

This study makes some novel contributions to the extant literature on platform governance. First, in contrast to a large body of platform literature dedicated to two-sided pricing strategies (Armstrong, 2006; Caillaud & Jullien, 2003; Parker & Van Alstyne, 2005; Rochet & Tirole, 2003), the issue of managing the quality of complementary applications has received only scant attention. As Boudreau and Hagiu (2009) note, "getting the price right" is not a sufficient condition to guarantee the success of a multisided market. Therefore, our work builds on Hagiu (2009a) and contributes directly to the discourse on the quality versus quantity trade-off in platform governance. Unlike Hagiu (2009a), who focuses solely on the exclusion strategy, we compare three different forms of strategy that have widespread adoption in the technology industry. Second, we add to emerging literature on first-party content (e.g., Hagiu & Spulber, 2013; Lee, 2013) by showing that a first-party applications strategy has important implications for platform governance, contributing to indirect network effects not only by increasing application variety but also by catering to consumers' quality preferences. However, while effective in improving application quality and platform profit, a first-party applications strategy is not always socially desirable. Third, although some prior studies have investigated the use of a subsidization strategy in a two-sided platform setting(Economides & Katsamakas, 2006; Eisenmann et al., 2006; Gawer & Cusumano, 2008; Lin et al., 2011; Parker & Van Alstyne, 2005), the primary consideration has been attracting initial adoption—that is, getting one side of the market on board to solve the chicken-and-egg dilemma when the platform is first launched (Caillaud & Jullien, 2003; Parker et al., 2016). We take one step further and examine this strategy from a quality regulation perspective. Therefore, we consider the strategy of selective subsidization conditional on quality level rather than indiscriminately subsidizing all players on one side of the market.

2 Related Literature

Our study is directly related to the literature on quality management in two-sided markets. Researchers have long recognized that the strength of indirect network effects depends not only on the variety of complementary goods but also on their quality (Kim et al., 2014). Earlier work has suggested that information asymmetry likely leads to certain types of market failure with suboptimal quality levels, and a minimum quality standard often results in social desirable outcomes (Akerlof, 1970; Leland, 1979). Ronnen (1991) further shows that a minimum quality standard strategy not only resolves the underprovision of quality but also reduces excessive quality differentiation, thereby improving social welfare even in the absence of network externalities.

Several studies have also examined the effect of exclusive distribution on content quality. For example, in a model in which two distributors bargain with a content producer for distribution rights, Stennek (2014) shows that exclusive distribution may encourage investments in quality and force the competitor to reduce its price, therefore benefiting all viewers. In the context of media platforms, D'Annunzio (2017) demonstrates that a content provider always prefers granting premium content exclusively to a single distribution platform; however, a vertically integrated content provider has lower incentive to invest in quality than an independent provider.

Some researchers have studied how open access on one side of the market influences quality provision. For example, Jeon and Rochet (2010) show that in an open access model, a for-profit journal tends to publish more low-quality articles to increase its profit from author fees. Surprisingly, quality degradation occurs even when the journal is not for-profit and aims to maximize readers' welfare. Casadesus-Masanell and Llanes (2015) compare incentives to invest in platform quality between open-source and proprietary platforms. They show that under certain conditions, an open platform may lead to higher investment than a proprietary platform.

The work that is most closely related to ours appears in the literature examining the trade-off between quantity and quality of complementary goods in a two-sided market (Hagiu, 2009a; Zheng & Kaiser, 2013). In particular, Hagiu (2009a) proposes a model in which users value the quality of complementary goods in addition to variety, and quality preference is incorporated into the indirect network effect. He concludes that the incentive to exclude low-quality complementors depends on the relative preference for quality versus for quantity and on the fraction of high-quality complementors. Building on Hagiu (2009a) framework, Zheng and Kaiser (2013) study the determination of an optimal quality threshold for limiting entry. Notably, in both studies, the focus is placed solely on the exclusion strategy.

3 The Benchmark Model

We consider a two-sided platform with indirect network effects, where one side of the market joins to offer applications or content that enhances the value of the platform and the other side joins to consume the applications or content. For the purpose of exposition, we call the former "developers" and the latter "consumers." The platform charges a fixed access fee, p_d, to developers and a fixed access fee, p_c, to consumers. This model setup can accommodate a wide range of applications, including digital platforms, such as online market intermediaries (e.g., HomeAdvisor), and nondigital platforms, such as a job fair. For simplicity, we assume that each developer offers only one application. The applications offered by developers differ vertically, with quality being either high or low. We assume that a fraction, $\lambda \in [0,1]$, of the developers are of high quality, $q_h > 0$, and that $1-\lambda$ of the developers are of low quality, q_l. Without loss of generality, we normalize q_l to 0. As is customary, we assume that the platform has better information than consumers regarding application quality (Hagiu, 2009a). The platform observes the quality of each developer, but consumers only observe the value of λ; that is, they are not able to tell the quality of a specific developer before joining the platform (Belleflamme & Peitz, 2019).

Consider the case in which n developers (n_h and n_l denote the number of high-quality and low-quality developers, respectively) and m consumers join the platform. Let $\overline{q}$ be the average quality level of the n developers on the platform. The utility of a consumer joining the platform is given by

$$V(\theta_j) = w + \alpha_c n + \mu\overline{q} + \beta\overline{q}n - p_c - \theta_j,$$

where w is the standalone base utility of joining the platform. Many platforms, such as computer or smartphone operating systems, offer basic functionalities from which consumers derive positive utility even without outside, complementary applications. To avoid trivial solutions, we assume that $w > 0$ throughout the paper. In addition, α_c is the indirect network effect parameter on the consumer side.

Consumer utility also depends on the quality of the applications on the platform; to this end, μ and β can be viewed as measures of consumers' preferences associated with average quality and total quality (note that $\overline{q}=n_h q_h/n$), respectively. Throughout the paper, we assume that $\mu \geq 0$ and $\beta \geq 0$. The relative importance of μ and β is likely to be platform-specific. For example, average quality on a platform will be more important when (1) consumers consume applications from most of the developers that join the platform or (2) it is difficult for a consumer to observe an application's quality before purchase such that the quality level of his or her consumption is subject to chance. In contrast, on platforms where consumers use only a small fraction of the applications due to either limited demand or abundant supply and where it is relatively easy to obtain quality information before purchase, consumers are usually concerned with total quality—that is, the number of popular, high-quality applications.

We note that most digital platforms, such as video game consoles, streaming services, mobile app markets, or marketplaces for web browser plug-ins, belong to the latter category, with an abundant supply of applications that far exceed consumers' demand and a variety of reputation systems by which consumers can tell high-quality applications from low-quality ones with relative ease. Therefore, consumer quality preference is, for the most part, determined by β rather than μ.

The parameter θ_j is a horizontal differentiation parameter (e.g., consumers' learning cost) that is uniformly distributed on $[0, \theta_c]$. A consumer with parameter θ_j will join the platform if $V\left(\theta_j\right) \geq 0$. With θ_j following a uniform distribution on $[0, \theta_c]$, the consumer-side demand function can be written as

$$m = \frac{w + \alpha_c n + \mu\overline{q} + \beta\overline{q}n - p_c}{\theta_c}. \tag{1}$$

Equivalently, the inverse demand function on the consumer side can be written as

$$p_c = w + \alpha_c n + \mu\overline{q} + \beta\overline{q}n - m\theta_c. \tag{2}$$

On the developer side, joining a platform with m consumers, the utilities of a high-quality developer and a low-quality developer are given by

$$U_h\left(\theta_i\right) = \alpha_{dh} m - bn - p_d - \theta_i,$$

and

$$U_l\left(\theta_i\right) = \alpha_{dl} m - bn - p_d - \theta_i,$$

where α_{dh} and α_{dl} are the indirect network effect parameters for the high- and low-quality developers, respectively. We assume that $\alpha_{dh} \geq \alpha_{dl}$, which implies that high-quality developers benefit more from the consumer network than low-quality developers do. For example, Amazon has created the "Amazon's Choice" badge, which has been used since 2015 to recommend highly rated, well-priced products that are ready to ship immediately, thereby directing more traffic to high-quality third-party sellers. The parameter b is the same-side network effect parameter. We assume that $b > 0$; that is, a negative network effect arises among developers because they prefer less competition (Eisenmann et al., 2006). The parameter θ_i is a horizontal differentiation parameter that represents the application development cost, and it is uniformly distributed on $[0, \theta_d]$.

The demand function on the developer side is

$$n = n_h + n_l = \frac{\overline{\alpha}_d m - p_d}{(\theta_d + b)}, \tag{3}$$

where $\overline{\alpha}_d = \lambda\alpha_{dh} + (1-\lambda)\alpha_{dl}$. The inverse demand function on the developer side is

$$p_d = \overline{\alpha}_d m - (\theta_d + b)n, \tag{4}$$

and the average quality is

$$\overline{q} = \frac{n_h q_h}{n} = \left(\lambda + \frac{\rho m}{n}\right) q_h,$$

where $\rho = \lambda(1-\lambda)\left(\alpha_{dh} - \alpha_{dl}\right)/\theta_d$.

We assume that the platform incurs an operating cost that is proportional to the overall network size; that is, with n developers and m consumers, the platform's operating cost is ηmn. Similar to Rochet and Tirole (2003), mn can be interpreted here as the volume of "transactions" between consumers and developers. Thus, we can write the platform's profit as

$$\Pi^0 = p_d n + p_c m - \eta mn,$$

where the first term is the total platform access fees collected from the n developers, the second term is the total access fees collected from m consumers, and the last term is the operating cost of the platform.

Substituting (2) and (4) into the profit function and collecting terms, we can formulate the platform's profit optimization problem as

$$\max_{m\geq 0,\ n\geq 0} \Pi^0 = (w + \lambda\mu q_h)\, m + \xi mn - \theta_c m^2 - (\theta_d + b)\, n^2 + \rho q_h m^2(\beta + \frac{\mu}{n}),$$

where $\xi = \overline{\alpha}_d + \alpha_c + \beta\lambda q_h - \eta$.

Assumption: The platform's profit optimization problems are jointly concave in the decision variables.

We make one general assumption throughout the paper: for each model that we study, the platform's optimization problem is well-defined; that is, its objective function is jointly concave in the decision variables. The assumptions to ensure joint concavity can be different under models with different quality regulation strategies, which are detailed separately in the online appendix. When we compare different strategies, we only consider parameter spaces that ensure joint concavity for all models under comparison.

We derive the optimal developer and consumer network sizes, the corresponding optimal developer and consumer access fees, and the optimal profit for the platform under different quality regulation

Table 1: Model Parameters

Parameter	Definition
m (n)	The number of consumers (developers) who join the platform
p_c (p_d)	Platform access fee for consumers (developers)
λ	The fraction of high-quality developers
θ_i (θ_j)	Developer (consumer) horizontal differentiation parameter (e.g., development cost on the developer side and learning cost on the consumer side)
θ_d (θ_c)	Upper bound of the developer (consumer) horizontal differentiation parameter
$\alpha_{dl}(\alpha_{dh})$	Developer-side *indirect* network effect parameter for low-quality (high-quality) developers
α_c	Consumer-side *indirect* network effect parameter
b	Developer-side *direct* network effect parameter
μ	Consumer preference parameter associated with average quality
β	Consumer preference parameter associated with total quality
q_h (q_l)	Quality level of high-quality (low-quality) developers
n_h (n_l)	The number of high-quality (low-quality) developers that join the platform
η	Platform operating cost parameter
w	Consumer standalone utility of joining the platform
U (V)	Developer (consumer) utility
Π	Platform profit

strategies in Appendix 1. To avoid uninteresting and trivial cases, we consider interior equilibria only. Table 1 presents a list of model parameters. Lemma 1.1 summarizes the equilibrium outcome in the benchmark model.

Lemma 1.1 (**Equilibrium Properties under the Benchmark**). *When the platform does not regulate application quality:*

(1) The equilibrium developer and consumer network sizes, m^{0*} and n^{0*}, are the unique solutions of the following two equations:

$$m^{0*} = \frac{w + \lambda\mu q_h + \ \xi n^{0*}}{2(\theta_c - \rho q_h(\beta + \frac{\mu}{n^{0*}}))},$$

and

$$n^{0*} = \sqrt{\frac{\mu\rho q_h m^{0*2}}{\xi m^{0*} - 2(\theta_d + b)n^{0*}}}.$$

(2) The network sizes m^{0*} and n^{0*} are both increasing in consumer quality preference parameters μ and β.

Proof: All proofs of the lemmas and propositions appear in Appendix 3.

As we discussed previously, on most technology platforms, such as video game consoles, mobile app markets, or music/video streaming services, application supply is abundant and consumers are primarily attracted to popular, high-quality applications. Therefore, they have a much stronger preference for the total quality than for the average quality of the applications. Because we are primarily interested in technology platforms in this work, for each quality regulation strategy, we consider the special case of $\mu = 0$, for which analytically tractable equilibria emerge.

Lemma 1.2 (**Equilibrium Properties under the Benchmark when $\boldsymbol{\mu = 0}$**). *When the platform does not regulate application quality and consumer quality preference depends only on total quality (i.e., $\mu = 0$):*

(1) The equilibrium developer and consumer network sizes are, respectively,

$$m^{0*} = \frac{2(\theta_d + b)w}{4(\theta_d + b)(\theta_c - \rho q_h) - \xi^2}$$

and

$$n^{0*} = \frac{w\xi}{4(\theta_d + b)(\theta_c - \rho q_h) - \xi^2} \ .$$

The corresponding equilibrium average quality is

$$\overline{q}^{0*} = [\lambda + \frac{2\rho\,(\theta_d + b)}{\xi}]q_h.$$

The equilibrium profit for the platform is

$$\Pi^{0*} = \frac{(\theta_d + b)w^2}{4(\theta_d + b)(\theta_c - \rho q_h) - \xi^2}.$$

(2) The equilibrium profit Π^{0*} and network sizes m^{0*} and n^{0*} are all increasing in consumer quality preference parameter β.

(3) While the equilibrium total quality, $n_h^{0*}q_h$, is increasing in consumer quality preference parameter β, the equilibrium average quality, $\overline{q}^{0*}$, is decreasing in β.

When quality preference is only a function of the total quality of the applications on the platform, we have closed-form solutions for the equilibria, as stated in lemma 1.2. Both equilibrium profit of the platform and equilibrium network sizes increase with the network effects ($\overline{\alpha}_d$ and α_c), the consumers' preference for total quality (β), and the fraction of high-quality developers (λ), but decrease in the operating cost coefficient (η), the developer direct network effect parameter (b), and the horizontal differentiation parameters of developers and consumers (θ_d and θ_c).

Interestingly, part (3) of the lemma suggests that when quality preference depends only on total quality, the platform admits more high-quality developers as β increases, but it also admits still more low-quality developers to take advantage of indirect network effects to attract more consumers. As a result, a higher β leads to greater total quality provision but, at the same time, lowers the average quality of the applications on the platform.

4 Quality Regulation Strategies

Because consumers derive greater utility with a higher quality of the applications, the platform is incentivized to implement quality regulation strategies to influence quality provision when such strategies lead to higher profits. We consider three widely used quality regulation strategies: exclusion, subsidization, and first-party applications. In this section, we characterize the equilibrium outcomes under each strategy and compare them with the benchmark model in which no quality regulation strategy is employed. We then present a comparison of the three quality regulation strategies in section 5.

4.1 Exclusion

With exclusion, the platform uses a quality threshold to exclude low-quality developers from joining the platform (Hagiu, 2009a; Zheng & Kaiser, 2013). In our model with two quality levels, the strategy dictates that only high-quality developers are granted access to the platform. As a result, the average quality of developers on the platform under exclusion is $\overline{q}^E = q_h$.

The developer utility function and the consumer utility function remain the same as those in the benchmark model. Because only high-quality developers are allowed access under exclusion, the demand function of the developer side is

$$n = n_h = \frac{\lambda(\alpha_{dh}m - p_d)}{(\theta_d + \lambda b)}, \tag{5}$$

and the inverse demand function of the developer side is

$$p_d = \alpha_{dh}m - \frac{n(\theta_d + \lambda b)}{\lambda}. \tag{6}$$

Using the average quality $\overline{q}^E = q_h$ under exclusion, we can write the demand function of the consumer side as

$$m = \frac{w + \alpha_c n + \mu q_h + \beta q_h n - p_c}{\theta_c}. \tag{7}$$

Comparing the demand functions (5) and (7) to those under the benchmark model, (3) and (1), we can see the trade-off under exclusion clearly. On the one hand, excluding low-quality developers raises the average quality of applications on the platform to q_h, which makes the platform more attractive to consumers, all else being equal. On the other hand, exclusion leads to a lower number of developers n, which reduces the attractiveness of the platform to consumers. Therefore, as prior literature has revealed (Hagiu, 2009a), an exclusion strategy is associated with a trade-off between quality and quantity. Depending on the relative strength of the two effects, the net effect of exclusion on consumer network size, m, can be either positive or negative.

The inverse demand function of the consumer side is

$$p_c = w + \alpha_c n + \mu q_h + \beta q_h n - m\theta_c. \tag{8}$$

The profit of the platform remains

$$\Pi^E = p_d n + p_c m - \eta m n.$$

Using (6) and (8), we can formulate the profit optimization problem for the platform as

$$\max_{m\geq 0,\ n\geq 0} \Pi^E = wm + \xi_1 mn + \mu q_h m - \theta_c m^2 - \frac{(\theta_d + \lambda b)}{\lambda} n^2,$$

where $\xi_1 = \alpha_{dh} + \alpha_c + \beta q_h - \eta$. Assuming that Π^E is jointly concave in m and n, proposition 1.1 characterizes the platform's equilibrium outcomes under exclusion.

Proposition 1.1 (**Equilibrium Properties under Exclusion**). *When the platform excludes low-quality developers:*

(1) The equilibrium developer and consumer network sizes are, respectively,

$$m^{E*} = \frac{2(\theta_d + \lambda b)(w + \mu q_h)}{4(\theta_d + \lambda b)\theta_c - \lambda {\xi_1}^2},$$

and

$$n^{E*} = \frac{\lambda \xi_1 (w + \mu q_h)}{4(\theta_d + \lambda b)\theta_c - \lambda {\xi_1}^2}.$$

The equilibrium profit for the platform is

$$\Pi^{E*} = \frac{(\theta_d + \lambda b){(w + \mu q_h)}^2}{4(\theta_d + \lambda b)\theta_c - \lambda {\xi_1}^2}.$$

(2) The equilibrium average quality $\overline{q}^{E*} = q_h$, which is greater than that of the benchmark model without quality regulation, $\overline{q}^{0*}$. The equilibrium profit of platform Π^{E*}, developer network size n^{E*}, and consumer network size m^{E*}all increase in consumers' preferences associated with average quality μ and total quality β.

Not surprisingly, exclusion always increases the average quality of applications on the platform. The platform's equilibrium profit under exclusion increases when consumer preferences for average quality μ and total quality β become higher, and so do equilibrium network sizes. In the special case of $\mu = 0$, we can derive more tractable observations of the effects of exclusion. The following proposition summarizes the properties of the exclusion strategy compared with the benchmark model without quality regulation when $\mu = 0$.

Proposition 1.2 (**Comparison between Exclusion and the Benchmark when $\boldsymbol{\mu = 0}$**). *When consumer quality preference depends only on total quality (i.e., $\mu = 0$) and the platform excludes the low-quality developers:*

(1) Exclusion does not always increase the platform's profit or the consumer network size. If there is a scarcity of high-quality developers (i.e., $\lambda < \lambda_0$,), exclusion leads to a lower platform profit and a smaller consumer network size, $\Pi^{E*} < \Pi^{0*}$and $m^{E*} < m^{0*}$.

(2) Under exclusion, it is possible that even high-quality developers are worse off, $U_h^{E*} < U_h^{0*}$, and the equilibrium number of high-quality developers is lower than that in the benchmark, $n^{E*} < n_h^{0*}$.

(3) *Exclusion improves platform profits, or* $\Pi^{E*} > \Pi^{0*}$*, when* $(\alpha_{dh} - \alpha_{dl})$ *is sufficiently large. When exclusion improves platform profits, it always leads to a larger consumer network,* $m^{E*} > m^{0*}$*, and higher access fees on the consumer side,* $p_c^{E*} > p_c^{0*}$*.*

According to part (1) of proposition 1.2, although exclusion always increases the average quality of applications on the platform, it does not necessarily result in a larger consumer network if the quality improvement is not sufficient to compensate for the reduced developer network size, resulting in lower consumer utility. Smaller consumer and developer network sizes would lead to lower platform profits. The condition in part (1) suggests that this is more likely to happen if the fraction of high-quality developers, λ, is sufficiently low. Under such a condition, exclusion would prevent a large fraction of developers from participating, significantly weakening the network size of the platform. This is also likely to happen when the difference in indirect network effects on the developer side $(\alpha_{dh} - \alpha_{dl})$ is small, the network effect on the consumer side (α_c) is high, and the operating cost (η) is low. Under these scenarios, consumers, developers, and the platform all prefer larger network sizes (or quantities) over higher quality. As a result, exclusion would not benefit the platform.

Part (2) of the proposition indicates that exclusion might not necessarily attract more high-quality developers to join the platform due to reduced consumer network size. When this happens, the network effects are so strong that quality improvement under exclusion is achieved at a great expense to the platform. As a result, exclusion significantly hurts the welfare of developers, regardless of their quality.

Part (3) suggests that when exclusion is beneficial to the platform, the platform charges a higher access fee to a larger consumer network to make a higher profit than it does in the benchmark model. In other words, when exclusion improves profits for the platform, the underlying mechanism is to build a smaller, elite developer network that allows the platform to profit from a larger number of consumers who have strong preferences for high quality.

4.2 Subsidization

In a two-sided market, subsidization has been shown to be particularly effective in boosting platform adoption and building market momentum (Gawer & Cusumano, 2008). To improve application quality in a platform ecosystem, the platform can subsidize high-quality developers to create incentives for them to participate. We consider a strategy under which the platform offers a subsidy of a fixed amount, $\gamma \geq 0$, to each high-quality developer that joins the platform. Such practices are becoming popular in platform markets: for example, when Uber launched in Seattle, to attract high-end ride providers, it subsidized town car participation by paying drivers even when they were not transporting customers.[5] By providing a subsidy to some developers but not others, the platform is able to implement a price discrimination strategy; that is, the platform can charge different access fees to high-quality and low-quality developers.

Under subsidization, the utility functions for low-quality developers and high-quality developers are

[5]See https://www.huffingtonpost.com/alex-moazed/7-strategies-for-solving-__b__6809384.html.

different because high-quality developers earn a subsidy, γ, which can be viewed as a quality premium. The utility function for low-quality developers remains unchanged as $U_l(\theta_i) = \alpha_{dl} m - bn - p_d - \theta_i$, and the utility function for high-quality developers becomes $U_h(\theta_i) = \alpha_{dh} m - bn + \gamma - p_d - \theta_i$. Given the utility functions, the number of low-quality developer joining the platform is $n_l = (1-\lambda)(\alpha_{dl} m - bn - p_d)/\theta_d$, and the number of high-quality developers joining is $n_h = \lambda(\alpha_{dh} m - bn + \gamma - p_d)/\theta_d$. Therefore, we can write the total number of developers in the market, $n = n_l + n_h$, as

$$n = \frac{\overline{\alpha}_d m - p_d + \gamma\lambda}{(\theta_d + b)}. \tag{9}$$

Comparing (9) to (3), all else being equal, we see that subsidization attracts more high-quality developers to join the platform, leading to a net change of $\gamma\lambda/(\theta_d + b)$ in the total number of developers. The inverse demand function on the developer side is

$$p_d = \overline{\alpha}_d m + \lambda\gamma - (\theta_d + b)n. \tag{10}$$

Comparing (10) to (4) in the benchmark model, we see that while the platform offers subsidy γ to high-quality developers, it also increases the access fee to low-quality developers by $\lambda\gamma$.

We calculate the average quality under subsidization as $\overline{q}^S = (n_h q_h + n_l q_l)/n$. Substituting n_h and n_l, we obtain

$$\overline{q}^S = \left(\lambda + \rho\frac{m}{n} + \frac{\gamma t}{n}\right) q_h, \tag{11}$$

where $t = \lambda(1-\lambda)/\theta_d$.

Therefore, compared with the benchmark case, the average quality under subsidization is increased by $\frac{\gamma t}{n} q_h$. A consumer's utility has the same form as in the benchmark model. We express the demand function on the consumer side as

$$m = \frac{w + \alpha_c n + \mu\overline{q} + \beta\overline{q}n - \ p_c}{\theta_c} + \frac{\gamma t q_h(\beta n + \mu)}{\theta_c n}. \tag{12}$$

Comparing (12) to (1), all else being equal, we see that the net change to the consumer network size from subsidization is $\gamma t q_h(\beta n + \mu)/(\theta_c n)$. The inverse demand function of consumers is

$$p_c = w + \alpha_c n + \mu\overline{q}^S + \beta\overline{q}^S n - m\theta_c. \tag{13}$$

We can write the profit of the platform under subsidization as

$$\Pi^S = p_d n + p_c m - \eta m n - \gamma n_h,$$

where γn_h is the total subsidy paid by the platform to high-quality developers. Using (10) and (13), we can formulate the profit optimization problem for the platform as

$$\max_{m\geq 0,\ n\geq 0,\ \gamma\geq 0} \Pi^S = (w+\lambda\mu q_h)\, m + \xi mn - \theta_c m^2 - (\theta_d + b)\, n^2$$
$$+m\left(\rho q_h m + \gamma t q_h\ \right)\left(\beta + \frac{\mu}{n}\right) - \gamma\rho m - \frac{\gamma^2\lambda(1-\lambda)}{\theta_d}.$$

We assume that Π^S is jointly concave in m, n, and γ and summarize the platform's equilibrium outcome under subsidization in proposition 2.1.

Proposition 2.1 (**Equilibrium Properties under Subsidization**). *When the platform offers a fixed subsidy $\gamma \geq 0$ to high-quality developers:*

(1) The equilibrium developer and consumer network sizes and subsidy (m^{S*}, n^{S*}, and γ^*) are the unique solutions of the following three equations:

$$m^{S*} = \frac{2\gamma^*\lambda(1-\lambda)}{\theta_d(t\beta q_h + \frac{tq_h\mu}{n^{S*}} - \rho)},$$

$$n^{S*} = \sqrt{\frac{\mu q_h m^{S*}}{2\gamma^* + ((\alpha_{dh}-\alpha_{dl}) - \beta q_h)m^{S*}}},$$

and

$$\gamma^* = \frac{(\mu q_h/n^{S*} + \beta q_h - (\alpha_{dh}-\alpha_{dl}))m^{S*}}{2}.$$

(2) Compared with the benchmark model without quality regulation, subsidization always increases the average developer quality, $\overline{q}^{S*} > \overline{q}^{0*}$.

(3) When $\alpha_{dh} - \alpha_{dl} > \mu q_h/n^{S*} + \beta q_h$, the optimal subsidy becomes zero.

The equilibrium network sizes are uniquely defined implicitly by the three equations in part (1) of proposition 2.1. By offering a subsidy, the platform will be able to attract more high-quality developers to join and, in turn, more consumers. However, the larger consumer network size could attract more low-quality developers to join as well. Therefore, the net effect of a subsidy on the average quality of the platform in equilibrium is not straightforward. Part (2) of proposition 2.1 states that the net effect of a subsidy on the average quality of the platform is always positive. Part (3) indicates that it is unnecessary to offer a subsidy to high-quality developers when the difference between the indirect network effect parameters on the developer side, $(\alpha_{dh} - \alpha_{dl})$, is high enough. In other words, when high-quality developers benefit from the consumer network significantly more than low-quality developers, such as when the endorsement of an "Amazon's choice" badge leads to significantly higher traffic to high-quality sellers, a subsidy based on quality will not improve profits.

When $\mu = 0$, we derive closed-form solutions for the equilibria under subsidization, which allows us to obtain more insight into subsidization in the rest of this subsection. Given the optimal subsidy, γ^*, and access fee to developers, p_d^{S*}, we define $p_{dh}^{S*} \triangleq p_d^{S*} - \gamma^*$ as the *effective* access fee charged to high-quality developers. We define $\tau = \lambda(1-\lambda)(\theta_d + b)(\beta q_h + \alpha_{dh} - \alpha_{dl})^2/\theta_d$. The following proposition characterizes the properties of γ^*, p_d^{S*}, and p_{dh}^{S*}, and Figure 1 illustrates those properties.

Proposition 2.2 (Equilibrium Properties under Subsidization when $\boldsymbol{\mu = 0}$). *When consumer quality preference depends only on total quality (i.e., $\mu = 0$) and the platform offers a fixed subsidy ($\gamma \geq 0$) to high-quality developers:*

(1) The optimal subsidy is:

$$\gamma^* = \frac{(\theta_d + b)w[\beta q_h - (\alpha_{dh} - \alpha_{dl})]}{4(\theta_d + b)\theta_c - \xi^2 - \tau}.$$

(2) While both the optimal subsidy, γ^*, and the optimal access fee to developers, p_d^{S*}, increase in consumer quality preference, β, the effective access fee to high-quality developers, p_{dh}^{S*}, decreases in consumer quality preference, β.

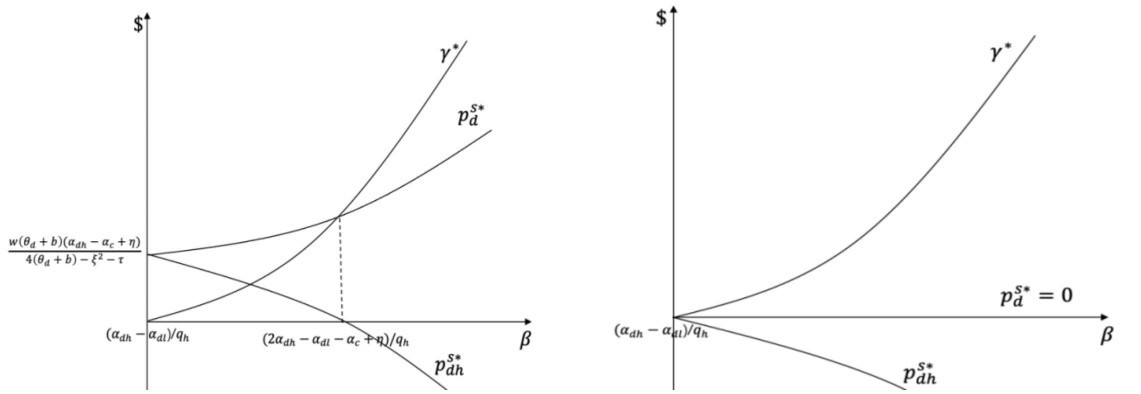
(3) The optimal subsidy decreases in $(\alpha_{dh} - \alpha_{dl})$. When $\alpha_{dh} - \alpha_{dl} > \beta q_h$, the optimal subsidy becomes zero.

(4) If consumer quality preference is sufficiently high (i.e., $\beta > (2\alpha_{dh} - \alpha_{dl} - \alpha_c + \eta)/q_h$), it is optimal for the platform to subsidize high-quality developers more than the optimal access fee so that they effectively get paid to join the platform (i.e., $\gamma^* > p_d^{S*}$ or $p_{dh}^{S*} < 0$).

In essence, subsidization is a form of price discrimination that allows the platform to charge differential access fees to developers according to their quality levels—specifically, p_d^{S*} to low-quality developers and $p_{dh}^{S*} = p_d^{S*} - \gamma^*$ to high-quality ones. Price discrimination enables the platform to achieve a desired average quality level and developer network size more efficiently than under uniform pricing. As consumers' quality preference, β, increases, the platform desires more high-quality developers to join but fewer low-quality developers, leading to a greater extent of price discrimination (as represented by γ^*). According to part (2) of proposition 2.2, the platform achieves its goal by increasing the access fee, p_d^{S*}, while simultaneously decreasing the *effective* access fee to high-quality developers, p_{dh}^{S*} (by making subsidy γ^* sufficiently large to offset p_d^{S*}). Increasing p_d^{S*} discourages low-quality developers that are not desirable to the platform. At the same time, with a sufficiently high subsidy, the reduced effective access fee for high-quality developers, p_{dh}^{S*}, attracts more of them to join.

Part (3) of proposition 2.2 suggests that instead of directly subsidizing high-quality developers, an alternative approach to improving application quality is to increase the difference between the indirect network effect parameters of high-quality and low-quality developers, $(\alpha_{dh} - \alpha_{dl})$, thus placing high-quality developers in a more advantageous position. Similar to proposition 2.1, when the difference is large enough, the platform does not need to provide a subsidy at all to attract high-quality developers.

High-quality developers benefit directly from the subsidization. When consumer quality preference, β, is high enough, as part (4) of proposition 2.2 shows, high-quality developers become so desirable that the platform is willing to offer such a high subsidy that the effective access fee to them becomes negative, and high-quality developers would actually get paid by the platform to join.



(a) $2\alpha_{dh} - \alpha_{dl} - \alpha_c + \eta > 0$

(b) $2\alpha_{dh} - \alpha_{dl} - \alpha_c + \eta \leq 0$

Figure 1: Illustrations of the Optimal Subsidy, γ^*, Access Fee p_d^{S*}, and Effective Access Fee for High-Quality Developers, p_{dh}^{S*}, as Functions of Consumer Quality Preference.

The following proposition compares the equilibrium outcomes between the subsidization strategy and the benchmark model.

Proposition 2.3 (Comparison between Subsidization and the Benchmark when $\mu = 0$). *When consumer quality preference depends only on total quality (i.e., $\mu = 0$):*

1. *Subsidization always increases platform profit, the developer network size, and the consumer network size (i.e., $\Pi^{S*} > \Pi^{0*}$, $n^{S*} > n^{0*}$, and $m^{S*} > m^{0*}$).*

2. *Subsidization leads to higher access fees for both low-quality developers and consumers (i.e., $p_d^{S*} > p_d^{0*}$, and $p_c^{S*} > p_c^{0*}$). Even with a subsidy, the effective access fee for high-quality developers can still be higher than under the benchmark model (i.e., $p_{dh}^{S*} > p_d^{0*}$).*

Proposition 2.3 demonstrates that subsidization is a powerful quality regulation strategy and uncovers the mechanisms through which it benefits the platform. According to part (1) of proposition 2.3 subsidization always improves the platform's profit. Intuitively, the benchmark model without quality regulation is a special case of subsidization, with $\gamma = 0$. Subsidization also leads to both a larger developer network size and a larger consumer network size for the platform.

With larger networks sizes, the platform can raise access fees for both sides. As we discuss in proposition 2.2, with a subsidy, the effective access fee for high-quality developers could even be negative. However, this does not necessarily happen all the time. As part (2) of proposition 2.3 indicates, there are cases in which even with a subsidy, the effective access fee for high-quality developers is higher than that in the benchmark model. In these cases, the platform would enjoy higher fees from larger network sizes, thereby increasing revenues. In summary, subsidization increases network size on both sides of the market and leads to higher average quality, thus allowing the platform to charge higher access fees, especially to consumers, and increase profits.

4.3 First-Party Applications

First-party applications, common in video game or video streaming service industries, provide a mechanism for platform owners to enter into content development and distribution. Such content or applications are usually exclusively distributed on the native platform and therefore add to the appeal of the platform via differentiation (Lee, 2013). We consider another strategic use of first-party applications: to improve average application quality, the platform can develop and offer high-quality first-party applications directly to consumers.[6] In our setting, unlike third-party developers, the platform can develop and offer multiple applications. We assume that the platform incurs a development cost of kx^2 for producing x high-quality first-party applications. We relax this assumption and consider alternative cost functions in the section of model extensions.

With x first-party applications, the total number of applications offered on the platform is $n + x$, and the total number of high-quality applications is $n_h + x$, where $n_h = \lambda n + \rho m$. The average quality of applications on the platform is

$$\overline{q}^F = \frac{(n_h + x)q_h}{n + x} = \frac{(\lambda n + x + \rho m)q_h}{n + x}. \tag{14}$$

Recall that in the benchmark model, $\overline{q}^0 = n_h q_h / n$. It is straightforward that $\overline{q}^F \geq \overline{q}^0$; that is, compared with the benchmark, a first-party applications strategy increases average quality.

Consumer utility with first-party applications is given by

$$V(\theta_j) = w + \alpha_c(n + x) + \mu \overline{q}^F + \beta \overline{q}^F (n + x) - p_c - \theta_j. \tag{15}$$

When we substitute (14) into (15), we get the demand function of the consumer side:

$$m = \frac{w + \alpha_c(n + x) + \frac{\mu(\lambda n + x + \rho m)q_h}{n + x} + \beta q_h(\lambda n + x + \rho m) - p_c}{\theta_c}. \tag{16}$$

Comparing (16) with (1), all else being equal, with x first-party applications, we see that the platform can attract more consumers than under the benchmark model. The inverse demand function of the consumer side is

$$p_c = w + \alpha_c(n + x) + \frac{\mu(\lambda n + x + \rho m)q_h}{n + x} + \beta q_h(\lambda n + x + \rho m) - m\theta_c. \tag{17}$$

The utility of a high-quality developer joining the platform with m consumers is given by

$$U_h(\theta_i) = \alpha_{dh} m - b(n + x) - p_d - \theta_i,$$

[6] Although it is possible for the platform to produce low-quality applications in addition to high-quality ones, it can be shown that it is never profitable for the platform to produce only low-quality applications. It can also be shown that the platform must first produce at least some number of high-quality applications before it starts to produce low-quality ones. Therefore, for simplicity, in this work, we focus on the parameter space where the platform only produces high-quality first-party applications.

and the utility of a low-quality developer joining the platform with m consumers is given by

$$U_l(\theta_i) = \alpha_{dl} m - b(n+x) - p_d - \theta_i.$$

Therefore, the demand function of the developer side is

$$n = \frac{\overline{\alpha}_d m - bx - p_d}{(\theta_d + b)}, \tag{18}$$

and the inverse demand function of the developer side is

$$p_d = \overline{\alpha}_d m - bx - (\theta_d + b)n. \tag{19}$$

Comparing (18) to (3) in the benchmark model, we see that the number of third-party developers joining the platform is reduced by $bx/(\theta_d + b)$ because they now face competition from the platform itself. However, due to a larger consumer network, m, the overall effect of first-party applications on developer network size can be positive or negative.

With $n + x$ applications on the platform, the operating cost of the platform becomes $\eta m(n + x)$. Thus, the platform's profit with x first-party applications is

$$\Pi^F = p_d n + p_c m - \eta m(n+x) - kx^2,$$

where kx^2 is the development cost for first-party applications.

Substituting (4) and (16) into Π^F, we can formulate the platform's profit optimization problem when it develops first-party applications as follows:

$$\begin{aligned}\max_{m\geq 0,\ n\geq 0,\ x\geq 0} \Pi^F = {} & (w + \lambda\mu q_h)\, m + \xi mn + (\alpha_c + \beta q_h - \eta)\, mx - m^2\theta_c \\ & -n^2(\theta_d + b) - bnx + \rho q_h m^2 \left(\beta + \frac{\mu}{n+x}\right) - \ kx^2.\end{aligned}$$

Assuming that Π^F is jointly concave in m, n, and x, we derive proposition 3.1.

Proposition 3.1 (**Equilibrium Properties under First-Party Applications**). *When the platform offers x first-party applications with a development cost of kx^2:*

(1) The equilibrium developer and consumer network sizes and the optimal number of first-party applications (m^{F*}, n^{F*}, and x^*) are the unique solutions of the following three equations:

$$m^{F*} = \frac{w + \lambda\mu\rho q_h + \xi n^{F*} + (\alpha_c + \beta q_h - \eta)x^{F*}}{2(\theta_c - \rho\beta q_h) - 2\mu\rho q_h(n^{F*} + x^{F*})},$$

$$n^{F*} = \frac{x^* - [bx^* - ((1-\lambda)\beta q_h - \eta - \overline{\alpha}_d)m^{F*}]/2k}{2k(2\theta_d + b)},$$

and

$$x^* = \frac{(\alpha_c + \beta q_h - \eta)\, m^{F*} - bn^{F*} - \mu\rho q_h (m^{F*})^2/(n^{F*})^2}{2k}.$$

(2) Compared with the benchmark model without quality regulation, a first-party applications strategy always increases the average developer quality, $\overline{q}^{F*} > \overline{q}^{0*}$.

Part (1) of proposition 3.1 characterizes the equilibrium outcomes of the platform under first-party applications. Part (2) confirms that with the introduction of high-quality first-party applications, the average quality of the applications on the platform is higher than in the benchmark model.

When $\mu = 0$, we can derive more granular insight about the first-party applications strategy. The following proposition illustrates some properties of the equilibrium under first-party applications. To simplify exposition, we define $\delta = [b^2\theta_c + (\theta_d + b)(\alpha_c + \beta q_h - \eta)^2 - b\xi(\alpha_c + \beta q_h - \eta) + \lambda(1 - \lambda)\beta q_h (\alpha_{dh} - \alpha_{dl})(4(\theta_d + b)k - b^2)/\theta_d]/k$. Figure 2 shows some of those properties visually.

Proposition 3.2 (Equilibrium Properties under First-Party Applications when $\boldsymbol{\mu = 0}$). *When consumer quality preference depends only on total quality (i.e.,* $\mu = 0$*) and the platform develops* x *first-party applications with a development cost of* kx^2*:*

(1) The optimal number of first-party applications the platform should develop is

$$x^* = \frac{w[(\alpha_c + \beta q_h - \eta)(\theta_d + b) - b\xi/2]}{k[4\theta_c(\theta_d + b) - \xi^2 - \delta]}.$$

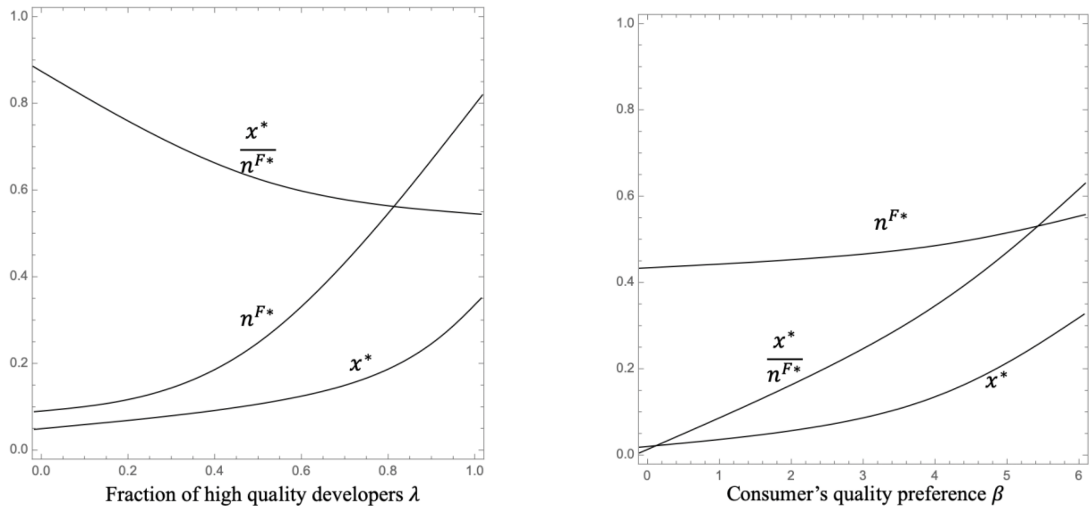
If b *is sufficiently high or* λ *is sufficiently high, it is not in the interest of the platform to develop first-party applications (i.e.,* $x^* = 0$*).*

(2) When k *is sufficiently low or* $1 - \lambda$ *is sufficiently high, the platform becomes a closed one; that is, no third-party developers will join the platform (i.e.,* $n^{F*} = 0$*).*

(3) When $x^* > 0$*, the optimal number of first-party applications,* x^**, is increasing in the fraction of the high-quality developers,* λ *(or equivalently, decreasing in* $[1 - \lambda]$*). However, the ratio of first-party applications to third-party applications,* x^*/n^{F*}*, always decreases with* λ*.*

(4) When $x^* > 0$*, the optimal number of first-party applications,* x^**, is increasing in consumer quality preference,* β*. In addition, the ratio of first-party applications to third-party applications,* x^*/n^{F*}*, is also increasing in* β*.*

Proposition 3.2 states that the platform will only offer first-party applications when the fraction of high-quality developers is smaller than a given threshold. In addition, if the consumer network effect, α_c, the consumer quality preference, β, or the value of high quality, q_h , is too low or if the platform operating cost, η, is too high, a first-party applications strategy is not preferred. Under these scenarios, the added value from more high-quality applications is low because the increase in the network effect (through both application variety and quality) is not strong enough to offset platform maintenance and



(a) (b)

Figure 2: The number of developers, n^{F*}, the number of first-party applications, x^*, and their ratio, x^*/n^{F*}, as functions of the fraction of high-quality developers, λ (shown in [a]), and as functions of consumer quality preference, β (shown in [b]). $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $\eta = 0.6$, $b = 1.5$, and $k = 27$ in both (a) and (b). $\beta = 2.7$ in (a), and $\lambda = 0.55$ in (b).

development costs. Otherwise, investments in developing first-party applications will lead to a higher platform profit.

Part (2) suggests that when the market is fraught with low-quality applications, the platform is more likely to offer a vertically integrated platform with no outside participation. This is because offering first-party applications will lead to greater externality as a large fraction of low-quality developers free-ride on the quality improvement. Therefore, the platform raises its access price to such a high level that third-party developers refrain from joining. Because the platform lacks the power of price discrimination, even high-quality developers are excluded. Consider the plight of the video game console maker Atari in the 1980s: our analyses suggest that closing the platform to outside participation might have been a sensible decision if it could produce first-party games at a reasonably low cost.

Intuitively, when the fraction of low-quality developers, $1 - \lambda$, increases, one would expect the platform to offer more first-party applications to compensate for the lower average quality. Surprisingly, part (3) of proposition 3.2 suggests the opposite: the platform would offer *fewer* first-party applications when the fraction of low-quality developers, $1 - \lambda$, increases. Note that with first-party applications, the platform is unable to completely internalize the development cost because it lacks the ability to price discriminate and must set a uniform access fee for both high-quality and low-quality developers. Charging a high access fee to developers will discourage high-quality developers and therefore weaken the effectiveness of quality improvement. However, charging a low access fee will allow more low-quality developers to enter and free-ride on the quality improvement (and the resulting larger consumer base) brought about by the first-party applications, thus diluting the effect of quality improvement. Therefore, when there is a large fraction of low-quality developers, greater externality deters the platform from creating more first-party applications. Part (3) also suggests that although the optimal number of first-party applications, x^*, is increasing in the fraction of high-quality developers, λ, the equilibrium number of third-party applications, n^{F*}, increases at a much faster rate. This is because as λ increases, application quality improvement comes from both more high-quality third-party developers (a first-order effect) and more first-party applications (a second-order effect through x^*), leading to higher incentives for developers to join. Thus, with a large λ, the platform is more likely to be dominated by third-party applications.

Part (4) indicates that when consumers have higher quality preferences, not surprisingly, the platform is willing to develop more first-party applications. With a high β, the platform is better able to recover a large part of the development costs from the consumer side, taking advantage of consumers' willingness to pay for quality. While the number of third-party developers, n^{F*}, also increases with consumer quality preferences, its rate of increase is lower than that of first-party applications because the average quality of third-party applications is lower than that of first-party applications (due to the presence of low-quality developers). Thus, with a high value of β, the platform is more likely to be dominated by first-party applications.

The following proposition compares the equilibrium outcomes between the first-party applications strategy and the benchmark model when $\mu = 0$.

Proposition 3.3 (**Comparison between First-Party Applications and the Benchmark when $\boldsymbol{\mu = 0}$**)**.** *When consumer quality preferences depend only on total quality (i.e., $\mu = 0$), a first-party applications strategy always increases platform profit, consumer network sizes, and the access fee to consumers (i.e., $\Pi^{F*} > \Pi^{0*}$, $m^{F*} > m^{0*}$, and $p_c^{F*} > p_c^{0*}$).*

Proposition 3.3 shows that a first-party applications strategy is also an effective quality regulation strategy for the platform. The benchmark model without quality regulation can be viewed as a special case of first-party applications, with $x = 0$. Therefore, it is not surprising that the platform will fare better with first-party applications. As average quality improves, the platform attracts more consumers, which makes the platform more appealing to developers through indirect network effects. However, the introduction of first-party applications also intensifies competition, thus reducing developer utility because of the negative same-side network effect. As a result, this strategy may not necessarily lead to a larger developer network size and, in some cases, may even exclude outside developer participation altogether. We also show that the platform is able to charge higher access fees and thus increase revenues. The increased revenues from access fees would be sufficient to offset development costs for first-party applications.

5 Optimal Quality Regulation Strategy

In this section, we investigate the platform's optimal choice of quality regulation strategy and discuss the relative advantages and limitations of the different strategies. Because we are primarily interested in digital platforms and because comparisons under $\mu > 0$ are intractable, we start with the case of $\mu = 0$ and then consider the generalizability of the findings under $\mu > 0$.

5.1 Optimal Quality Regulation Strategy under μ= 0

We derive the results presented in this subsection under the assumption that $\mu = 0$. To simplify exposition, we will not repeat it hereafter.

5.1.1 Exclusion versus Subsidization

In proposition 4, we characterize the platform's choice between exclusion and subsidization.

Proposition 4 (Comparison between Exclusion and Subsidization). *Comparing the strategies of subsidization and exclusion:*

(1) Subsidization is the dominant choice over exclusion (i.e., $\Pi^{S*} \geq \Pi^{E*}$). In fact, exclusion is a special case of subsidization; that is, for every optimal exclusion strategy, there always exists an equivalent subsidization strategy.

(2) Exclusion achieves higher average application quality than subsidization (i.e., $\overline{q}^{E*} \geq \overline{q}^{S*}$).

(3) Compared with exclusion, subsidization leads to larger network sizes on both the developer and the consumer sides and higher access fees for developers (i.e., $n_h^{S*} > n^{E*}$, $m^{S*} > m^{E*}$, $p_d^{S*} > p_d^{E*}$).

Proposition 4 states that exclusion is dominated by subsidization as a quality regulation strategy because it is a special case of subsidization. With subsidization, the platform can always set the developer access fee, p_d^S, sufficiently high so that no low-quality developer finds it profitable to join the platform and then adjust the amount of subsidy γ accordingly to offset the high access fee to attract the desired amount of high-quality developers to join, thus achieving the same effect as exclusion. Therefore, subsidization is a more general and flexible quality regulation strategy than exclusion.

However, as we state in part (2) of proposition 4, an advantage of exclusion is that it achieves higher average quality than subsidization (higher than first-party applications as well). Recall that the average quality under exclusion is $\overline{q}^{E*} = q_h$, which is the highest average quality level that a platform can possibly achieve in our model setting. However, the highest quality level is not always desirable to a platform, which explains why subsidization dominates exclusion: with subsidization, the platform can balance application quantity and quality, while exclusion is a more rigid strategy with a constant quality level. Exclusion does have some appeal: If the objective is to achieve a high (or the highest, as in our model) average quality, exclusion is a more effective and direct strategy that is simpler to implement than subsidization and first-party applications. This might explain why exclusion is commonly used in practice, although it is not necessarily the profit-optimizing strategy. There may be other reasons beyond our model that contribute to exclusion being widely adopted in practice. For example, in our model, we focus on "goods" produced by outside developers rather than "bads" that may be harmful to consumers and whose presence on the platform may cause disutility.

Except for average quality, subsidization dominates almost every other aspect according to part (3) of proposition 4. Subsidization leads to larger network sizes on both developer and consumer sides and allows the platform to charge higher fees to developers, thereby generating sufficient revenue to offset the cost of subsidization and earn a higher profit.

Because exclusion is a special case of the subsidization strategy, the platform's optimal choice of quality regulation strategy is between subsidization and first-party applications. We turn to this choice next.

5.1.2 Optimal Choice of Quality Regulation Strategy

The following proposition characterizes the platform's optimal quality regulation strategy.[7]

Proposition 5 (**Comparison between Subsidization and First-Party Applications**). *The platform's optimal choice of a quality regulation strategy between subsidization and first-party applications can be characterized as:*

(1) If the first-party application development cost coefficient, k, is sufficiently low, a first-party applications strategy is optimal (i.e., $\Pi^{F*} > \Pi^{S*}$*).*

(2) Otherwise, there are two thresholds, $0 < \underline{\lambda} < \overline{\lambda} < 1$ *(defined in the proof), such that the subsidization strategy is optimal (i.e.,* $\Pi^{S*} > \Pi^{F*}$*) when* $\underline{\lambda} < \lambda < \overline{\lambda}$ *while the first-party applications strategy is optimal (i.e.,* $\Pi^{F*} > \Pi^{S*}$*) when* $0 < \lambda < \underline{\lambda}$ *or* $\overline{\lambda} < \lambda < 1$*.*

Intuitively, when the first-party applications development cost is sufficiently low, first-party applications should be the optimal choice for the platform, which is confirmed in part (1) of proposition 5. Surprisingly, we find that even when first-party application development costs are high, there are conditions under which first-party applications may still outperform subsidization. Part (2) of proposition 5 shows that this happens when the fraction of high-quality developers, λ, is either sufficiently low or sufficiently high. The reason for this is that when λ is either low or high, subsidization may not work effectively to achieve the desired average quality level (and therefore network sizes). When λ is too low, there are simply not enough high-quality developers for the platform to subsidize to improve average quality level without sacrificing developer network size significantly (recall that under subsidization, the platform also raises the access fee for low-quality developers). When λ is too high, the cost of subsidization could become substantial, and the subsidization strategy achieves only limited improvement in average quality because most developers that join the platform are high-quality anyway. In contrast, a first-party applications strategy does not suffer from these limitations, because the number of first-party applications to offer is fully under the discretion of the platform. Thus, in these situations, developing first-party applications is the strategy of choice for the platform. Conversely, when λ is moderate, the condition is just right for subsidization to fully leverage the power of price discrimination, thus making subsidization the optimal strategy.

Figure 3 illustrates the platform's optimal choice between subsidization and first-party applications graphically on the plane of the first-party development cost, *k,* and the fraction of high-quality developers,

[7] When we compare the first-party applications strategy with other strategies, we focus on the more interesting case in which the platform is still open with a positive number of developers joining (i.e., $k > b(\alpha_c + \beta q_h - \eta)/2\xi$).

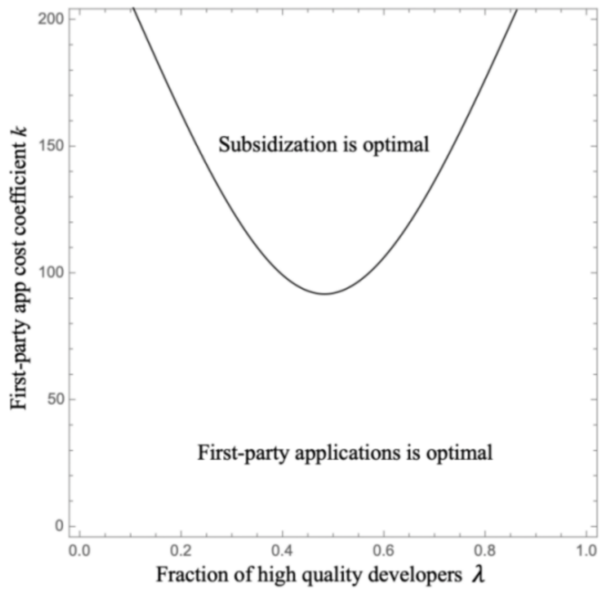


Figure 3: The Platform's Optimal Quality Regulation Strategy. $\beta = 2.7$, $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $b = 1.5$, and $\eta = 0.6$.

λ. The following proposition provides further insight into the platform's optimal quality regulation strategy.

Proposition 6 (Network Sizes and Quality Level under Optimal Strategy). *When the platform chooses between subsidization and first-party applications, the optimal strategy does not necessarily lead to either a higher average quality or a larger developer network; however, the optimal strategy always leads to a larger consumer network.*

Proposition 6 reveals that the platform prefers a quality regulation strategy (between subsidization and first-party applications) that enables it to grow the network size on the consumer side rather than achieving the highest average quality or the largest developer network. In other words, with quality regulation, the platform's ultimate goal is to become a larger platform with more consumers so that it can charge a higher access fee to consumers and improve its profitability. In Figure 4, we illustrate the platform's optimal quality regulation strategy when its objective is to maximize average application quality and contrast this choice with the profit-maximizing choice described in proposition 6.

5.2 Optimal Quality Regulation Strategy under $\mu > 0$

Comparisons of quality regulation strategies become analytically intractable when $\mu > 0$, because we do not have explicit characterizations of the equilibrium in most cases. Therefore, we conduct the comparisons numerically in this subsection to probe the robustness of our prior discoveries.

Figure 5 illustrates how μ affects the platform's optimal choice between subsidization and first-party applications. The curve for $\mu = 0$ in Figure 5 corresponds to the curve in Figure 3. The two curves for $\mu = 1$ and $\mu = 2$ demonstrate how μ potentially affects the platform's optimal choice.

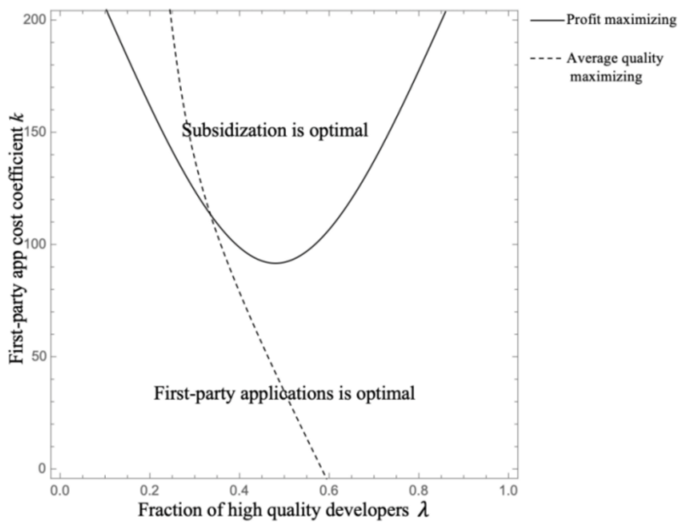


Figure 4: Profit-Maximizing Strategy versus Average Quality-Maximizing Strategy. $\beta = 2.7$, $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $b = 1.5$, and $\eta = 0.6$.

First, the different μ values in Figure 5 have similar curvatures. This implies that the structure of the platform's optimal choice between subsidization and first-party applications as characterized in proposition 5 is likely to hold for higher values of μ as well. Second, as μ becomes larger (i.e., as consumers care more about the average quality), the platform is more likely to choose subsidization as its quality regulation strategy over first-party applications. This is because subsidization is more effective at improving the average quality on the platform than first-party applications. To improve average quality, the platform must attract more high-quality developers while controlling the number of low-quality developers. As we have discussed, subsidization can achieve both goals effectively using two different access fees, whereas a first-party applications strategy suffers from free-riding by low-quality developers. Therefore, as μ becomes larger, subsidization will be more attractive to the platform than first-party applications.

6 Social Welfare Analyses

We have studied how different quality regulation strategies can be employed to improve a platform's profit. We now shift our attention to understanding their impacts on social welfare. Similar to the approach we adopted in the analyses of optimal quality strategy, we start with the case of $\mu = 0$ and then present a numerical study under $\mu > 0$ to verify the robustness of the findings.

6.1 Social Welfare under $\mu = 0$

For each of the models $t \in \{0, E, S, F\}$, social welfare is the sum of total consumer utility, V^*, total developer utility, U^*, and platform profit, Π^{t*}, defined as

$$W^{t*} = \int_0^n U(\theta_i)^{t*} di + \int_0^m V(\theta_j)^{t*} dj + \Pi^{t*}.$$

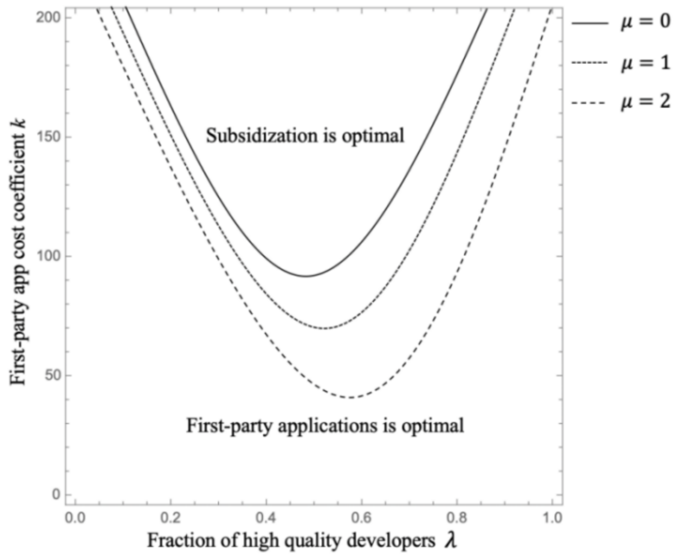


Figure 5: The Platform's Optimal Quality Regulation Strategy under Different Values of μ. $\beta = 2.7$, $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $b = 1.5$, and $\eta = 0.6$.

We detail the equilibrium social welfare under different quality strategies—benchmark W^{0*}, exclusion W^{E*}, subsidization W^{S*}, and first-party applications W^{F*}—in Appendix 1. We summarize the properties of social welfare under the different strategies in proposition 7.

Proposition 7 (**Social Welfare under Optimal Strategy**).

(1) *Subsidization always improves both platform profit and social welfare (i.e., $\Pi^{S*} \geq \Pi^{0*}$ and $W^{S*} \geq W^{0*}$). While a first-party applications strategy always improves platform profit (i.e., $\Pi^{F*} \geq \Pi^{0*}$), it does not necessarily improve social welfare. Exclusion does not necessarily improve either.*

(2) *When subsidization is the optimal strategy for the platform (i.e., $\Pi^{S*} \geq \Pi^{F*}$), it always leads to higher social welfare than first-party applications ($W^{S*} > W^{F*}$). However, the opposite is not necessarily true. Therefore, a social planner would choose subsidization over first-party applications more often than the platform would.*

In the previous section, we discussed how exclusion does not necessarily improve platform profit, while both subsidization and first-party applications do. According to part (1) of proposition 7, subsidization also improves social welfare because subsidization entails an internal transfer between the platform and high-quality developers. In contrast, first-party applications might not always increase social welfare because first-party applications introduce frictions. A profit-maximizing platform may have the incentive to overinvest in first-party applications even when it is not as efficient as third-party developers, which hurts social welfare. Recall that exclusion reduces network sizes, especially developer network size, increases the consumer access fee, and can lower platform profits, which are all detrimental to social welfare.

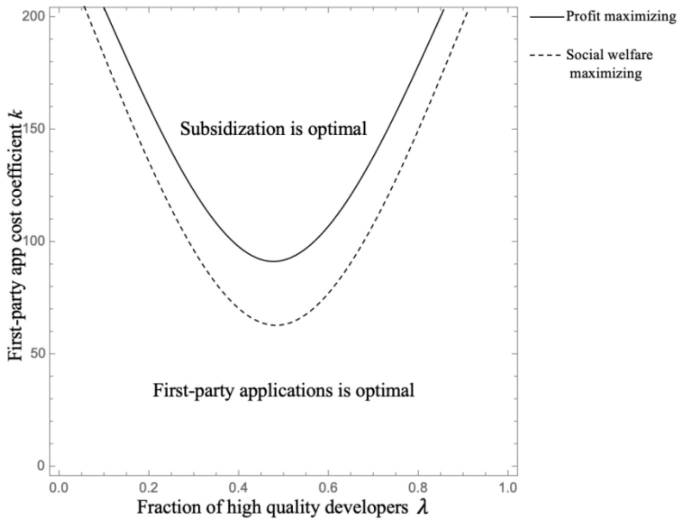


Figure 6: Profit-Maximizing Quality Regulation Strategy versus Social Welfare–Maximizing Quality Regulation Strategy. $\beta = 2.7$, $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $b = 1.5$, and $\eta = 0.6$.

Part (2) of the proposition suggests that subsidization, in addition to improving profit, is also the most social welfare–friendly quality regulation strategy among the three. Indeed, when subsidization is optimal, or when it maximizes platform profits, it always maximizes social welfare. However, when a first-party applications strategy is optimal for the platform, it may not maximize social welfare, which suggests that a welfare-maximizing social planner will prefer subsidization more frequently than the platform. Figure 6 shows the difference between the platform's choice and the social planner's choice between the two strategies. The area under which subsidization is optimal is larger for the social planner and subsumes that for the platform, implying that the social planner is more likely than the platform to choose subsidization as the optimal quality regulation strategy.

6.2 Social Welfare under μ> *0*

Because there is no analytically tractable solution when $\mu > 0$, we conduct a numerical study to understand the extent to which the findings we discovered under $\mu = 0$ will hold. We show how changing levels of μ affect the social welfare–maximizing quality regulation strategy between subsidization and first-party applications in Figure 7. Comparing the curves with different μ values in the figure, we find that observations derived in proposition 7 do not change significantly as μ becomes larger.

7 Extensions

We further investigate how changes in some model parameters and assumptions may affect the findings we obtained in previous sections. Specifically, we study the case in which developers with different quality levels may have different application development costs and the scenario in which the platform has a concave first-party application development cost instead of a convex one. We focus on the analytically tractable scenario of $\mu = 0$ throughout this section. The equilibrium outcomes of these

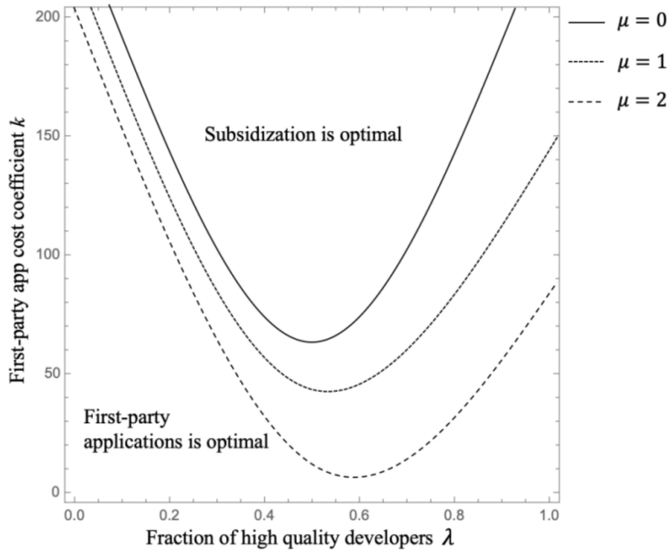


Figure 7: Social Welfare–Maximizing Quality Regulation Strategy under Different Values of μ. $\beta = 2.7$, $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $b = 1.5$, and $\eta = 0.6$.

extensions, along with the assumptions required to ensure that the platform's problems are well-defined, appear in Appendix 2. For the sake of brevity, in this section, we only document results that are new or different from the baseline models.

7.1 Heterogenous Application Development Cost

It is typical for development costs to increase with the quality level of an application. We relax the assumption that both types of developers have the same form of development cost and consider a case similar to that in Hagiu (2009a), where the development cost function takes the form of $C(q)\theta_i$, and

$$C(q) = \begin{cases} c & if\ q = q_h \\ 1 & if\ q = q_l \end{cases}.$$

We assume that $c > 1$ so that high-quality developers incur a higher cost. This also leads to different utility functions for developers with different quality levels. Specifically, joining a platform with m consumers, the utility of a high-quality developer is

$$U_h(\theta_i) = \alpha_{dh} m - bn - p_d - c\theta_i,$$

and the utility of a low-quality developer stays the same as the one in earlier models:

$$U_l(\theta_i) = \alpha_{dl} m - bn - p_d - \theta_i.$$

We solve the models with these utility functions and derive the optimal decisions and equilibrium outcomes for all quality regulation strategies in Appendix 2. The following proposition summarizes the impact of heterogenous development cost.

Proposition 8 (**Heterogeneous Development Cost**). *Consider the case in which high-quality developers incur a development cost, $c\theta_i$, where $c > 1$:*

(1) Under subsidization, the optimal subsidy decreases in c. Under first-party applications, the optimal number of first-party applications is not monotonic in c; however, when c is sufficiently high, the platform becomes a closed one; that is, no third-party developers will join the platform.

(2) In each quality regulation model $t \in \{0, E, S, F\}$, the optimal platform profit, as well as consumer and developer network sizes, are all decreasing in c.

Heterogenous development costs have little structural impact on the exclusion strategy because c is merely a constant scalar on high-quality developers' development costs. However, due to high-quality developers' cost disadvantage, they are less likely to join the platform compared with the benchmark model, all else being equal. As c increases, fewer high-quality developers will join the platform in the absence of platform intervention. Although providing a higher subsidy helps attract them to the platform, it is more costly and significantly hurts developer network size (recall that the platform has to raise access prices for low-quality developers at the same time when it subsidizes high-quality developers). As a result of the trade-off between quantity and quality, the platform reduces the amount of subsidy offered to high-quality developers when c is higher. Most of the structural results under subsidization we presented in section 4.2 carry over. Under a first-party applications strategy, when the development cost parameter for high-quality developers is sufficiently high, the platform would choose to rely completely on first-party applications because it has a cost advantage over high-quality developers, and allowing low-quality developers access would lead to their free-riding on the quality improvement. As a result, the platform raises the access price high enough to exclude outside participation altogether.

Higher development costs for high-quality developers hurt the platform in general. As part (2) of proposition 8 suggests, it leads to smaller developer and consumer network sizes and reduces platform profit. Its impact on the platform's optimal choice of quality regulation strategy between subsidization and first-party applications, however, is more subtle and does not strictly make either strategy more attractive. Figure 8 illustrates the change in the platform's optimal choices of strategy for different values of c.

7.2 Concave First-Party Application Development Cost

We assumed a convex development cost for first-party applications in section 4.3. However, application development may exhibit economies of scale because the platform can leverage existing human capital and physical assets required for application development or improve efficiency as more applications are developed due to learning (Banker & Kemerer, 1989). We consider a concave first-party application development cost function in which development cost takes the form of $k\sqrt{x}$ for producing x first-party

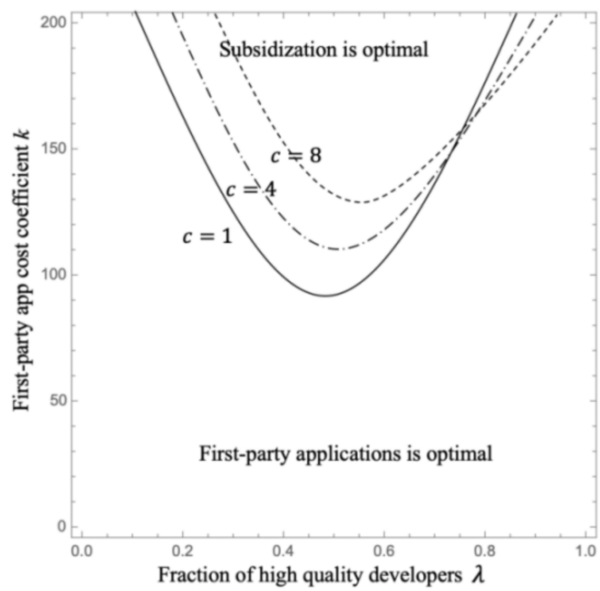


Figure 8: The Platform's Optimal Quality Regulation Strategy when $c = 1$, $c = 4$, and $c = 8$. $\beta = 2.7$, $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $b = 1.5$, and $\eta = 0.6$.

applications. The following proposition compares the properties of the optimal number of first-party applications under the two cost functions.

Proposition 9 (**Concave First-Party Development Cost**). *When the first-party application development cost is $k\sqrt{x}$:*

(1) The optimal number of first-party applications decreases in the fraction of high-quality developers, λ. In contrast, it increases in λ when the development cost is kx^2.

(2) All else being equal, the optimal number of first-party applications under cost function $k\sqrt{x}$ is higher than that under cost function kx^2.

When the fraction of high-quality developers, λ, increases, it leads to two countervailing effects on the optimal number of first-party applications, x^*. On the one hand, the platform is better able to internalize quality improvement and recover development costs because fewer low-quality developers will free-ride, which increases the return on developing more first-party applications. On the other hand, a larger fraction of high-quality developers puts a limit on the quality improvement that can be achieved by developing first-party applications. When the development cost is kx^2, as we discussed in proposition 3.2, the former factor dominants the latter because, with a convex cost, the development cost goes up quickly with more first-party applications. The optimal number of first-party applications, x^*, would therefore increase as λ increases because the return on developing more first-party applications becomes stronger. In contrast, with economies of scale under the cost function $k\sqrt{x}$, the development cost is less of a concern because it increases more slowly as more first-party applications are developed. Therefore, the latter effect of increasing λ dominates the former. When λ is low, the platform can afford to develop more first-party applications to achieve significant quality improvement without incurring too high of a

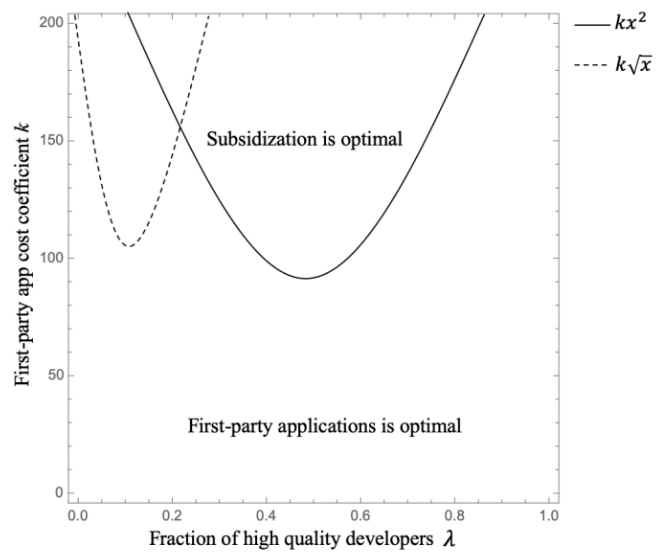


Figure 9: The Platform's Optimal Quality Regulation Strategy under Different First-Party Development Costs. $\beta = 2.7$, $q_h = 0.4$, $\theta_c = \theta_d = 2$, $\alpha_c = \alpha_{dh} = 0.6$, $\alpha_{dl} = 0.5$, $w = 1$, $b = 1.5$, and $\eta = 0.6$.

cost due to economies of scale. As λ increases, the marginal quality improvement that can be achieved by developing first-party applications becomes weaker. As a result, the platform will gradually reduce the number of the first-party applications. Part (2) of the proposition suggests that with economies of scale, as expected, the concave cost function, $k\sqrt{x}$, leads to a higher optimal number of first-party applications than the convex cost function, kx^2.

Changing the cost function from convex to concave does not necessarily make either subsidization or a first-party applications strategy more attractive. Figure 9 provide an example to illustrate the shift in choice of strategies. With cost $k\sqrt{x}$, subsidization becomes more attractive than first-party applications when λ is small, whereas a first-party applications strategy becomes more favorable when λ is large.

8 Discussion and Conclusions

With platforms becoming an increasingly popular business model in the technology industry, the role of a platform company transitions from coordinating internal economic activities to also providing boundary resources to outside complementors and regulating the conduct of firms within its platform ecosystem (Boudreau & Hagiu, 2009). While prior literature has provided many insights into the pricing strategies in a two-sided market (Hagiu, 2006, 2009b; Jeon & Rochet, 2010), regulation of the quality of complementary applications has received little research attention. In this study, we bridge this gap and compare three strategies that are widely employed in practice: excluding access to low-quality complementors, providing a subsidy to high-quality complementors, and developing high-quality first-party applications. Our analyses reveal that it is imperative for platforms to understand the mechanisms underlying the quality regulation strategies because, under a wide range of scenarios, implementing one of these strategies will lead to higher platform profits and often result in greater social

Table 2: Summary of Findings ($\mu = 0$)

	Exclusion	Subsidization	First-party applications
Advantages	Achieves the highest quality; easy to implement	Price discriminates developers according to quality	Particularly useful when there is a scarcity of high-quality developers or when the platform has a development cost advantage
Limitations	Its rigidity sometimes leads to lower profits and smaller networks	Becomes inefficient when λ is either too large or too small	Causes free-riding by low-quality developers; may lead to a closed platform
Compared with benchmark (i.e., without quality regulation)	Does not necessarily improve profit or network size	Improves profits and network sizes of both developers and consumers	Improves profits and consumer network size
Equilibrium average quality	Highest	Higher than the benchmark	Higher than the benchmark
When is the policy optimal?	Never, dominated by subsidization	When k is high and the distribution of high- and low-quality developers is more even	When k is low or when k is high and the distribution of high- and low-quality developers is more lopsided
Implications for social welfare	Does not always improve social welfare	Always improves social welfare	Does not always improve social welfare

welfare. Interestingly, strategies aimed at increasing application variety and those aimed at improving application quality need not be in conflict, as prior research has suggested (Hagiu, 2009a); instead, both objectives can be achieved simultaneously if the platform makes smart choices. To highlight the insights, in Table 2, we provide a summary of our major discoveries.

We show that each of the three strategies has unique advantages and limitations. Under exclusion, a platform can achieve a high quality level with relatively straightforward implementation. However, being the least flexible among the three, exclusion does not necessarily improve either platform profit or social welfare. In contrast, providing a subsidy to high-quality developers improves both because of its power of price discrimination, and therefore subsidization is a particularly attractive choice if the platform faces high first-party development costs. However, subsidization becomes increasingly ineffective if the platform is fraught with low-quality developers, and it is not cost efficient when third-party developers are predominantly of high quality. Under these conditions, a first-party applications strategy works particularly well if platform development costs are sufficiently low, but such a strategy may suffer from the overprovision of first-party applications to the extent that it may hurt social welfare and lead to a vertically integrated platform that excludes outside participation. In addition, under a first-party applications strategy, the platform faces the challenge of internalizing development costs due to the free-riding of low-quality developers. This issue is most prominent when high-quality developers and low-quality ones are about equal in quantity.

Our research also reveals several important managerial implications. For example, although the strategy of exclusion appears intuitively appealing, it may lead to undesirable consequences in certain contexts, and therefore its adoption should be carefully weighed against other alternatives. In contrast, platform designs that involve subsidizing high-quality complementors (e.g., the actions taken by Google's Android platform) or setting differential platform access fees based on application quality can often make the platform more profitable and socially desirable at the same time. Moreover, as many platforms (e.g., Netflix) start investing aggressively in the development of exclusive first-party applications, managers need to carefully evaluate whether choosing such a strategy is advantageous, taking into consideration factors such as cost efficiency relative to outside developers and quality distribution among third-party applications. Our study provides concrete guidelines to help managers make these decisions. For example, we show that when the platform can enhance the indirect network effect for high-quality developers (e.g., by directing more transactions to them through a recommender system), the need to subsidize high-quality developers is greatly reduced.

Several limitations of our model provide opportunities for further research. First, while we assume that third-party quality provision is exogenously determined, another useful quality regulation strategy might incentivize low-quality producers to exert greater effort and improve the quality of their applications, which will endogenize the application development stage. Second, one reason that many platforms deny access to some third-party developers is to exclude harmful applications, whose presence on the platform leads to negative network effects. Considering both economic "goods" and "bads" in the same model could lead to a more complete understanding of the comparisons between the different quality regulation strategies. Finally, a study that considers a combination of several quality regulation strategies, such as the use of both exclusion and first-party applications at the same time, could provide deeper insights into how limitations of a single strategy can be remedied.

References

Akerlof, G. A. (1970). The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, *84*(3), 488–500.

Armstrong, M. (2006). Competition in two-sided markets. *RAND Journal of Economics*, *37*(3), 668–691.

Banker, R. D., & Kemerer, C. F. (1989). Scale economies in new software development. *IEEE Transactions on Software Engineering*, *15*(10), 1199–1205.

Belleflamme, P., & Peitz, M. (2019). Managing competition on a two-sided platform. *Journal of Economics & Management Strategy*, *28*(1), 5–22.

Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs. devolving control. *Management Science*, *56*(10), 1849–1872.

Boudreau, K., & Hagiu, A. (2009). Platform rules: Multi-sided platforms as regulators. In A. Gawer (Ed.), *Platforms, markets and innovation* (pp. 163–191). Edward Elgar Publishing.

Caillaud, B., & Jullien, B. (2003). Chicken & egg: Competition among intermediation service providers. *RAND Journal of Economics*, *34*(2), 309–328.

Casadesus-Masanell, R., & Llanes, G. (2015). Investment incentives in open-source and proprietary two-sided platforms. *Journal of Economics and Management Strategy*, *24*(2), 306–324.

Ceccagnoli, M., Forman, C., Huang, P., & Wu, D. J. (2012). Cocreation of value in a platform ecosystem: The case of enterprise software. *MIS Quarterly*, *36*(1), 263–290.

D'Annunzio, A. (2017). Vertical integration in the TV market: Exclusive provision and program quality. *International Journal of Industrial Organization*, *53*(July), 114–144.

Economides, N., & Katsamakas, E. (2006). Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science*, *52*(7), 1057–1071.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2006). Strategies for two-sided markets. *Harvard Business Review*, *84*(10), 92–101.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2009). Opening platforms: How, when and why? In A. Gawer (Ed.), *Platforms, markets and innovation* (pp. 131–162). Edward Elgar Publishing.

Gawer, A., & Cusumano, M. A. (2002). *Platform leadership: How intel, microsoft, and cisco drive industry innovation*. Harvard Business School Press.

Gawer, A., & Cusumano, M. A. (2008). How companies become platform leaders. *MIT Sloan Management Review*, *49*(2), 28–35.

Gawer, A., & Henderson, R. (2007). Platform owner entry and innovation in complementary markets: Evidence from Intel. *Journal of Economics & Management Strategy*, *16*(1), 1–34.

Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: The boundary resources model. *Information Systems Journal*, *23*(2), 173–192.

Hagiu, A. (2006). Pricing and commitment by two-sided platforms. *RAND Journal of Economics*, *37*(3), 720–737.

Hagiu, A. (2009a). *Quantity vs. quality and exclusion by two-sided platforms* (Unpublished Work).

Hagiu, A. (2009b). Two-sided platforms: Product variaty and pricing structures. *Journal of Economics & Management Strategy*, *18*(4), 1011–1043.

Hagiu, A. (2014). Strategic decisions for multisided platforms. *MIT Sloan Management Review*, *55*(2), 71–80.

Hagiu, A., & Spulber, D. (2013). First-party content and coordination in two-sided markets. *Management Science*, *59*(4), 933–949.

Huang, P., Ceccagnoli, M., Forman, C., & Wu, D. (2013). Appropriability mechanisms and the platform partnership decision: Evidence from enterprise software. *Management Science*, *59*(1), 102–121.

Jeon, D.-S., & Rochet, J.-C. (2010). The pricing of academic journals: A two-sided market perspective. *American Economic Journal: Microeconomics*, *2*(2), 222–255.

Kim, J.-H., Prince, J., & Qiu, C. (2014). Indirect network effects and the quality dimension: A look at the gaming industry. *International Journal of Industrial Organization*, *37*(November), 99–108.

Lee, R. S. (2013). Vertical integration and exclusivity in platform and two-sided markets. *American Economic Review*, *103*(7), 2960–3000.

Leland, H. E. (1979). Quacks, lemons, and licensing: A theory of minimum quality standards. *Journal of Political Economy*, *87*(6), 1328–1346.

Lin, M., Li, S., & Whinston, A. B. (2011). Innovation and price competition in a two-sided market. *Journal of Management Information Systems*, *28*(2), 171–202.

Parker, G., & Van Alstyne, M. (2005). Two-sided network effects: A theory of information product design. *Management Science*, *51*(10), 1494–1504.

Parker, G., & Van Alstyne, M. (2017). Innovation, openness, and platform control. *Management Science*, *64*(7), 3015–3032.

Parker, G., Van Alstyne, M., & Choudary, S. P. (2016). *Platform revolution: How networked markets are transforming the economyand how to make them work for you.* WW Norton & Company.

Rochet, J.-C., & Tirole, J. (2003). Platform competition in two-sided markets. *Journal of the European Economic Association*, *1*(4), 990–1029.

Ronnen, U. (1991). Minimum quality standards, fixed costs, and competition. *The RAND Journal of Economics*, *22*(4), 490–504.

Song, P., Xue, L., Rai, A., & Zhang, C. (2018). The ecosystem of software platform: A study of asymmetric cross-side network effects and platform governance. *MIS Quarterly*, *42*(1), 121–142.

Stennek, J. (2014). Exclusive quality—why exclusive distribution may benefit the TV-viewers. *Information Economics and Policy*, *26*(March), 42–57.

Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, *21*(4), 675–687.

Zheng, Y., & Kaiser, H. (2013). Optimal quality threshold of admission in a two-sided farmers' market. *Applied Economics*, *45*(23), 3360–3369.