# Quality Regulation on Two-Sided Platforms: Exclusion, Subsidization, and First-party Application

**Peng Huang**
**Robert. H. Smith School of Business, University of Maryland**

**Gaoyan Lyu**
**Peking University**

**Yi Xu**
**Robert. H. Smith School of Business, University of Maryland**

## 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 application strategies make the platform owner better off, with greater profits, higher average quality, and a larger consumer network size, but only subsidization always improves social welfare. In addition, the tradeoff between subsidization and first-party application strategies depends on the development cost of first-party applications, as well as the fraction of high-quality complementors, but the relationship is not monotonic. Interestingly, our results demonstrate that the platform does not have to sacrifice application quantity for higher application quality. With the right choice, 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 the 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 and 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 and 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 and Henfridsson 2013), or managing intellectual properties within the ecosystem (Huang et al. 2013, Parker and 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 and Henderson 2007, Hagiu 2014, Parker and Van Alstyne 2005, Song et al. 2018, Tiwana et al. 2010).

Despite progress, one of the understudied but fundamentally important questions that remain in platform governance is the regulation of the quality of complementary applications (Hagiu 2009a). The importance of quality regulation is highlighted by the collapse of the videogame market in the early 1980s, where unrestricted entry resulted in a market for "lemons" flooded by poor-quality games, leading to the bankruptcy of over 90 percent of videogame developers as well as the failure of the dominant videogame platform at the time, Atari (Boudreau and Hagiu 2009). In contrast, the later success of Nintendo was partly attributed to its restrictive platform access strategy in which it 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 the access to low-quality complementors is still being widely adopted (such as 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 fbFund in

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

partnership with venture capitalists that awarded seed grants to selected startups dedicated to developing Facebook applications.[2] In addition, many two-sided platform owners, such as manufacturers of video game consoles (e.g., PlayStation or Xbox) and media streaming service providers (e.g., Netflix, Hulu or 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, hit applications, sometimes offered as part of a product bundle, play an important role in attracting an initial critical mass of platform adopters as well as winning the battle with competing platforms, especially when third-party applications are subject to multihoming (Hagiu and Spulber 2013, Rochet and Tirole 2003).

Although some scholars have started to tackle the issue of quality regulation on two-sided platforms with network effects, research in this area thus far has focused primarily on the strategy of exclusion based on a quality threshold (Hagiu 2009a, Zheng and 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 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 its 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 amount of subsidy to high-quality complementors, and 3) it employs the option of producing high-quality, first-party applications/content at a cost and therefore improves the average quality of applications in the platform ecosystem.

Our analyses reveal several important observations. First, we show that the widely adopted exclusion strategy is a special case of the subsidization strategy, i.e., for every optimal exclusion strategy there is an

---

[2] 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 web TV series *House of Cards* by Netflix, *The Handmaid's Tale* by Hulu, and the movie *Manchester by the Sea* by Amazon Prime Video.

[4] In this work we use application variety and application quantity interchangeably.

equivalent subsidization strategy that achieves the same level of profit. However, there exist conditions under which exclusion is strictly dominated by subsidization strategy, which is more flexible due to its mechanism of price discrimination. Second, compared to the benchmark scenario without platform owner intervention, both subsidization and first-party application strategies make the platform owner better off, with greater profits, higher average quality, a larger consumer network and a higher consumer access fees, but only subsidization always improves social welfare. An important insight is that, in contrast to the exclusion strategy (Hagiu 2009a), the adoption of the other two strategies does not require sacrificing quantity in order to improve quality (or vice versa); in fact, both strategies can achieve greater quantity *and* quality of applications at the same time. Third, the tradeoff between subsidization and first-party application strategies depends on the development cost of first-party applications, as well as 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 application, the platform faces difficulty in internalizing the development cost, primarily due to the freeriding 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 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 cost 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 also gain some additional insights.

This study makes a few 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, Bernard and Jullien 2003, Parker and Van Alstyne 2005, Rochet and Tirole 2003), the issue of managing the quality of complementary applications has received only scant attention. As Boudreau and Hagiu (2009) noted, ‘getting the price right’ is not a sufficient condition that guarantees the success of a multi-sided market. Therefore, our work builds on Hagiu (2009a) and contributes directly to the discourse on the quality vs. quantity tradeoff in platform governance. Different from Hagiu (2009a) that focuses solely on the strategy of exclusion, we compare three different forms of strategy that have seen widespread adoption in the technology industry. Second, we also add to an emerging literature on first-party content (e.g., Hagiu and Spulber 2013, Lee 2013) by showing that such a 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, first-party application strategy is not always socially desirable. Third, although some prior studies investigated the use of subsidization strategy in a two-sided platform setting (Economides and Katsamakas 2006, Eisenmann et al. 2006, Gawer and Cusumano 2008, Lin et al. 2011, Parker and Van Alstyne 2005), the primary consideration was on attracting initial adoption; i.e., getting one side of the market on board so as to solve the chicken-and-egg dilemma when the platform is first launched (Bernard and Jullien 2003, Parker et al. 2016). We take one step further and examine this strategy from a quality regulation perspective. Therefore, the strategy we consider is one of selective subsidization conditional on quality level, instead of 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 suggests 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 under-provision of quality but also reduces excessive quality differentiation, therefore improves social welfare even in the absence of network externalities.

A number of studies also examined the effect of exclusive distribution on content quality. For example, in a model where 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 price, therefore benefiting all viewers. Under 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 incentives to invest in quality than an independent one.

Some researchers have studied the effect of open access on one side of the market on quality provision. For example, Jeon and Rochet (2010) shows that under an open access model, a for-profit journal tends to publish more low-quality articles in order to increase 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, and opening up one side of a proprietary platform may lower incentives to invest in platform quality.

The work that most closely relates to ours is the stream of literature that studies the tradeoff between quantity and quality of complementary goods in a two-sided market (Hagiu 2009a, Zheng and Kaiser 2013). In particular, Hagiu (2009a) proposes a model where users value the quality of complementary goods in addition to their variety, and the 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 vs. for quantity, and on the fraction of high-quality complementors. Building on his framework, Zheng and Kaiser (2013) study the determination of 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 join to offer their applications or content that enhance the value of the platform, and the other side join 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 a developer, and a fixed access fee $p_c$ to a consumer. Such a model setup can accommodate a wide range of applications, including digital platforms such as online market intermediaries (e.g., HomeAdvisor) as well as non-digital 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 $(1-\lambda)$ of the developers are of low quality $q_l$. Without loss of generality, we normalize $q_l$ to 0. As customary, we assume that the platform has superior information than consumers regarding application quality (Hagiu 2009a): The platform observes the quality of each developer, but consumers only observe the value of $\lambda$, i.e., they are not able to tell the quality of a specific developer prior to joining the platform (Belleflamme and Peitz 2019).

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

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

$w$ is the stand-alone base utility of joining the platform. Many platforms, such as computer or smartphone operating systems, offer basic functionalities from which consumers derive positive utilities even without outside, complementary applications. To avoid trivial solutions, we assume $w > 0$ throughout the paper. $\alpha_c$ is the indirect network effect parameter on the consumer side. Consumer utility also depends on the quality of the applications on the platform: $\mu$ and $\beta$ can be viewed as measures of consumers' preferences associated with average quality and total quality (note that $\bar{q}$=$n_h q_h/n$), respectively. We assume $\mu \geq 0$ and $\beta \geq 0$ throughout the paper. The relative importance of $\mu$ and $\beta$ is likely platform-specific: for example, the average quality will be more important on a platform where 1) consumers consume applications from most developers that join the platform, or 2) it is difficult for a consumer to observe an application's quality prior to purchase such that the quality level of her consumption is subject to chance. In contrast, on platforms where consumers consume only a small fraction of applications due to either limited demand or abundant supply, and where it is relatively easy to obtain quality information prior to purchase, consumers are usually concerned with total quality, i.e., the number of high-quality, hit applications. We note that most digital platforms, such as video game consoles, streaming services, mobile app markets, or marketplaces for web browser plugins, belong to the latter category, with abundant supply of applications that far exceed a consumer's demand and a variety of reputation systems by which consumers can tell high-quality applications apart from low-quality ones with relative ease, and therefore consumer quality preference is for the most part determined by $\beta$ rather than $\mu$. $\theta_j$ is a horizontal differentiation parameter – such as learning cost on the part of consumers – that is uniformly distributed on $[0, \theta_c]$.

A consumer with parameter $\theta_j$ will join the platform if $V(\theta_j) \geq 0$. With $\theta_j$ following a uniform distribution on $[0, \theta_c]$, the demand function of the consumer side can be written as:

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

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

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

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:

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

and

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

where $\alpha_{dh}$ and $\alpha_{dl}$ are the indirect network effect parameters for the high- and low-quality developers, respectively. We assume $\alpha_{dh} \geq \alpha_{dl}$ 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 since 2015 to recommend highly rated, well-priced products ready to ship immediately, directing more traffic to high-quality third-party sellers. $b$ is same-side network effect parameter. We assume that $b > 0$; that is, negative network effect arises among developers because they prefer less competition (Eisenmann et al. 2006). $\theta_i$ 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:

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

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

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

and the average quality is:

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

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

We assume that the platform incurs an operating cost that is proportional to the overall network size, i.e., with $n$ developers and $m$ consumers, the platform's operating cost is $\eta mn$. Similar to Rochet and Tirole (2003), here $mn$ can be interpreted as the volume of "transactions" between consumers and developers. Hence, the profit of the platform can be written 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 above profit function and collecting terms, the platform's profit optimization problem can be formulated 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 we define $\xi = \bar{\alpha}_d + \alpha_c + \beta\lambda q_h - \eta$.

**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 in the paper, 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 stated separately in detail 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 policies in Appendix 1. To avoid uninteresting and trivial cases, throughout the paper, we consider interior equilibria only. Table 1 presents a list of model parameters. Lemma 1.1 summarizes the equilibrium outcome in the benchmark model.

**Table 1. Model Parameters**

| **Parameter** | **Definition** |
|---|---|
| $m$ ($n$) | The number of consumers (developers) that join the platform |
| $p_c$ ($p_d$) | Platform access fee for consumers (developers) |
| $\lambda$ | The fraction of high-quality developers |
| $\theta_i$ ($\theta_j$) | Developer (consumer) horizontal differentiation parameter, e.g., development cost on the developer side and learning cost on the consumer side |
| $\theta_d$ ($\theta_c$) | Upper bound of the developer (consumer) horizontal differentiation parameter |
| $\alpha_{dl}(\alpha_{dh})$ | Developer side *indirect* network effect parameter for low (high) quality developers |
| $\alpha_c$ | Consumer side *indirect* network effect parameter |
| $b$ | Developer side *direct* network effect parameter |
| $\mu$ | Consumer preference parameter associated with average quality |
| $\beta$ | 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 |
| $\eta$ | Platform operating cost parameter |
| $w$ | Consumer stand-alone utility of joining the platform |
| $U$ ($V$) | Developer (consumer) utility |
| $\Pi$ | Platform profit |

***Lemma 1.1: Equilibrium Properties under 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* $\mu$ *and* $\beta$.

As discussed earlier, 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 high-quality, hit applications, and therefore have a much stronger preference for total quality than the average quality of applications. Because we are primarily interested in technology platforms in this work, for each quality regulation strategy, we consider the special case where $\mu$=0, for which analytically tractable equilibria emerge.

***Lemma 1.2: Equilibrium Properties under Benchmark ($\mu$=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:*

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

*and*

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

*respectively. The corresponding equilibrium average quality is:*

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

*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 $\Pi^{0*}$, and network sizes $m^{0*}$ and $n^{0*}$ are all increasing in consumer quality preference parameter $\beta$.*

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

**Proof:** All proofs of the lemmas and propositions are presented in Appendix 3.

When quality preference is only a function of total quality of applications on the platform, we have closed-form solutions for the equilibria as stated in Lemma 1.2. It is not difficult to see that both equilibrium profit of the platform and equilibrium network sizes increase in the network effects $\bar{\alpha}_d$ and $\alpha_c$, the consumers' preference on total quality $\beta$, and the fraction of high-quality developers $\lambda$, but decrease in the operating cost coefficient $\eta$, the developer direct network effect parameter $b$ and the horizontal differentiation parameters of developers and consumers $\theta_d$ and $\theta_c$.

Interestingly, part (3) of the lemma suggests that when quality preference depends only on total quality, the platform admits more high-quality developers as $\beta$ increases, but it also admits a greater number of low-quality developers in order to take advantage of indirect network effects to attract more consumers. As a result, a higher $\beta$ 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 higher quality of the applications on the platform, the platform has incentives to implement quality regulation strategies to influence quality provision when such strategies lead to a higher profit. We consider three widely used quality regulation strategies: exclusion, subsidization, and first-party application. In this section, we will characterize the equilibrium outcomes under each strategy, and compare them with the *benchmark* model where no quality regulation strategy is employed. The comparison among the three quality regulation strategies will be presented in the next section.

### 4.1 Exclusion

With exclusion, the platform uses a quality threshold to exclude low-quality developers from joining the platform (Hagiu 2009a, Zheng and 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 $\bar{q}^E = q_h$.

The developer utility function and the consumer utility function remain the same as the ones 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 $\bar{q}^E = q_h$ under exclusion, the demand function of the consumer side can be written 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 the ones 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, everything 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 revealed in earlier literature (Hagiu 2009a), the exclusion policy is associated with a tradeoff 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), the profit optimization problem for the platform can be formulated 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 we define $\xi_1 = \alpha_{dh} + \alpha_c + \beta q_h - \eta$. Assuming that $\Pi^E$ 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:*

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

*and*

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

*respectively. The equilibrium profit for 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* $\bar{q}^{E*} = q_h$*, which is greater than that of the benchmark model without quality regulation,* $\bar{q}^{0*}$*. The equilibrium profit of platform* $\Pi^{E*}$*, developer network size* $n^{E*}$*, and consumer network size* $m^{E*}$ *all increase in consumers' preferences associated with average quality* $\mu$ *and total quality* $\beta$*.*

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 $\mu$ and total quality $\beta$ become higher, so do equilibrium network sizes. In the special case where $\mu = 0$, we can derive more tractable observations on the effects of exclusion. The following proposition summarizes the properties of the exclusion strategy as compared to the benchmark model without quality regulation when $\mu = 0$.

***Proposition 1.2: Comparison between Exclusion and Benchmark ($\mu$=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. Particularly, 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,* $U_h^{E*} < U_h^{0*}$*, with the equilibrium number of high-quality developer lower than that in the benchmark,* $n^{E*} < n_h^{0*}$*;*

*(3) Exclusion improves the profit of the platform, or* $\Pi^{E*} > \Pi^{0*}$*, when* $(\alpha_{dh} - \alpha_{dl})$ *is sufficiently large. When exclusion improves the profit of the platform, it always leads to a larger consumer network* $m^{E*} > m^{0*}$ *and higher access fee on 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 doesn't necessarily result in a larger consumer network if the quality improvement is not sufficient to compensate for the reduced developer network size, resulting in a lower consumer utility. Smaller consumer and developer network sizes would lead to lower profit for the platform. The condition in part (1) suggests that this is more likely to happen if the percentage of high-quality developers, $\lambda$, is sufficiently low. Under such a condition, exclusion would prevent a large fraction of developers from participating, which significantly weakens the network sizes 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 effects on consumer side $\alpha_c$ is high, and the operating cost $\eta$ is low. Under these scenarios, consumers, developers, as well as 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 even 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 hurt 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 higher profit than it does in the benchmark model. In other words, when exclusion is profit improving for the platform, the underlying mechanism is to build a smaller, elite developer network which allows the platform to profit from a larger number of consumers who have strong preference for higher 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 and Cusumano 2008). To improve application quality in the 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 who 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 weren't transporting customers.[5] Note that by providing

[5] See https://www.huffingtonpost.com/alex-moazed/7-strategies-for-solving-_b_6809384.html.

a subsidy to some developers but not others, the platform is able to implement a price discrimination strategy; i.e., subsidization allows the platform to 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 different because high-quality developers earn a subsidy $\gamma$, 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, the total number of developers in the market $n = n_l + n_h$ can be written as

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

Comparing (9) to (3), with everything else being equal, we can 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 of the developer side is:

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

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

The average quality under subsidization can be calculated as $\bar{q}^S = (n_h q_h + n_l q_l)/n$. Substituting $n_h$ and $n_l$, we obtain

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

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

Therefore, compared to the benchmark case, the average quality under subsidization is increased by $\frac{\gamma t}{n} q_h$. The utility of a consumer has the same form as in the benchmark model. The demand function on the consumer side can be expressed as

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

Comparing (12) to (1), with everything else being equal, the net change on the consumer network size caused by subsidization is $\gamma(1-\lambda)\lambda q_h(\beta n+\mu)/(\theta_d\theta_c n)$. The inverse demand function of consumers is:

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

The profit of the platform under subsidization can be written as

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

where the last term is total subsidy paid by the platform to high-quality developers. Using (10) and (13), the profit optimization problem for the platform can be formulated as

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

Assuming that $\Pi^S$ is jointly concave in $m$, $n$, and $\gamma$, proposition 2.1 summarizes the platform's equilibrium outcome under subsidization.

***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 $\gamma^*$ are the unique solutions of the following three equations:*

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

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

*and*

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

*(2) As compared to the benchmark model without quality regulation, subsidization always increases the average developer quality, $\bar{q}^{S*} > \bar{q}^{0*}$.*

*(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 attracting more consumers. However, the larger consumer network size could attract more low-quality developers to join as well. Therefore, the net effect of 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 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 much more than low-quality developers, such as when the endorsement of "Amazon's choice" badge leads to significantly higher traffic to high-quality sellers, a subsidy based on quality will not be profit improving.

When $\mu = 0$, we derive closed-form solutions for the equilibria under subsidization which allow us to obtain more insight on subsidization in the rest of this section. Given the optimal subsidy $\gamma^*$ 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. 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 $\gamma^*$, $p_d^{S*}$, and $p_{dh}^{S*}$, and Figure 1 illustrates the properties.

***Proposition 2.2: Equilibrium Properties under Subsidization (μ=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, $\gamma^*$, and the optimal access fee to developers, $p_d^{S*}$, are increasing in consumer quality preference $\beta$, the effective access fee to high-quality developers, $p_{dh}^{S*}$ is decreasing in consumer quality preference $\beta$;*

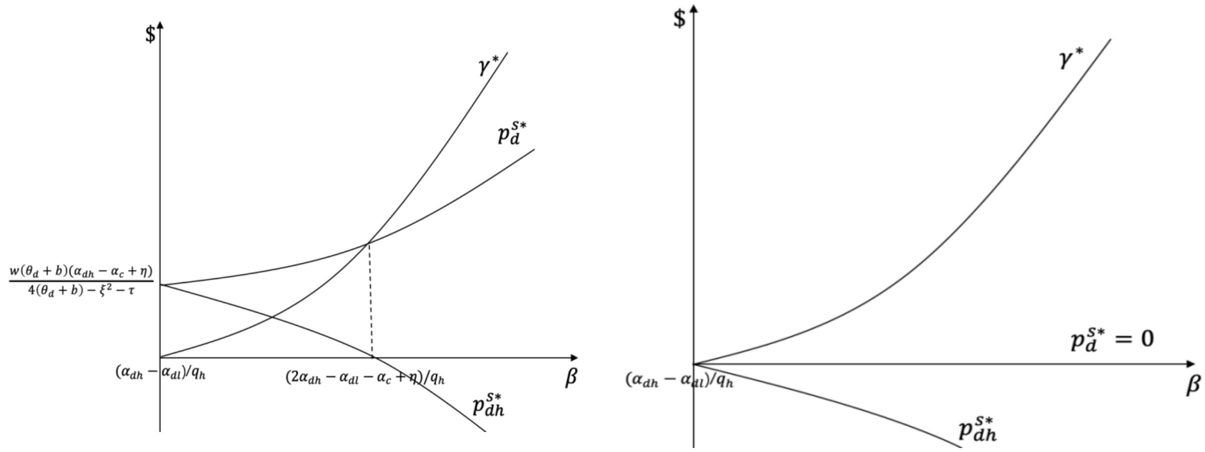
*(3) The optimal subsidy is decreasing in $(\alpha_{dh} - \alpha_{dl})$. When $\alpha_{dh} - \alpha_{dl} > \beta q_h$, the optimal subsidy becomes zero.*

*(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 which 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. The price discrimination enables the platform to achieve desired average quality level and network size on the developer side more efficiently than under uniform pricing. As consumers' quality preference $\beta$ increases, the platform desires more high-quality developers but less low-quality developers to join, leading to a greater extent of price discrimination (as represented by $\gamma^*$). 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 the high-quality developers $p_{dh}^{S*}$ (by making the subsidy $\gamma^*$ sufficiently large to offset $p_d^{S*}$). Increasing $p_d^{S*}$ discourages low-quality developers who are not desirable to the platform. At the same time, with a sufficiently high subsidy, the reduced effective access fee to high-quality developers, $p_{dh}^{S*}$, attracts more of them to join.

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

High-quality developers benefit directly from the subsidization. When consumer quality preference $\beta$ 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 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 $\gamma^*$, 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 parameters between the subsidization strategy and the benchmark model.

***Proposition 2.3: Comparison between Subsidization and Benchmark ($\mu$=0)***

*When consumer quality preference depends only on total quality, i.e., $\mu = 0$:*

*(1) Subsidization always increases the 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, $p_d^{S*} > p_d^{0*}$, $p_c^{S*} > p_c^{0*}$; Even with 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 profit of the platform. Intuitively, the benchmark model without quality regulation is a special case of subsidization with $\gamma = 0$. Subsidization also leads to both larger developer network size and larger consumer network size for the platform.

With bigger networks sizes, the platform can raise access fees to both sides. As we discuss in proposition 2.2, with the 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 where even with subsidy, the effective access fee to high-quality developers is higher than the one in the benchmark model. In these cases, the platform would enjoy higher fees on bigger network sizes, thereby much higher revenues. In summary, subsidization increases network size on both sides of the market and leads to higher average quality so that higher access fees, especially to consumers, can be charged to improve profit for the platform.

### 4.3 First-party Application

First-party applications, often seen in the video game or the video streaming service industries, provide a mechanism for platform providers to integrate into content development and publishing. Such content or applications are usually exclusively distributed on the native platform and therefore add to the appeal of the platform by creating differentiation (Lee 2013). Here, we consider another strategic use of first-party applications; i.e., 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 if it wants to. We assume that the platform incurs a development cost of $kx^2$ for producing $x$ high-quality, first-party applications. We will 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 given as

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

Recall that in the benchmark model, $\bar{q}^0 = n_h q_h / n$. It is straightforward to see that $\bar{q}^F \geq \bar{q}^0$; that is, compared to the benchmark, first-party application increases the average quality.

The consumer utility with first-party application is given by:

---

[6] Although it's 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.

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

Substitute (14) into (15), we get the demand function of the consumer side as

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

Comparing (16) with (1), everything else being equal, with *x* first-party applications, 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:

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

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:

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

and the inverse demand function of the developer side is:

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

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

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

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

where the last term is the development cost for the first-party applications.

Substituting (4) and (16) into $\Pi^F$, the platform's profit optimization problem, when it develops first-party applications, can be formulated as:

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

Assuming that $\Pi^F$ is jointly concave in $m$, $n$, and $x$, we have the following proposition.

***Proposition 3.1: Equilibrium Properties under First-party Application***

*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:*

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

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

*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) As compared to the benchmark model without quality regulation, first-party application always increases the average developer quality, $\bar{q}^{F*} > \bar{q}^{0*}$.*

Part (1) of proposition 3.1 characterizes the equilibrium outcomes of the platform under first-party application. 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 that in the benchmark model.

When $\mu = 0$, we can derive more granular insight about the first-party application 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 these properties visually.

***Proposition 3.2: Equilibrium Properties under First-party Application (μ=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* $\lambda$ *is sufficiently high, then 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,* $n^{F*} = 0$*;*

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

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

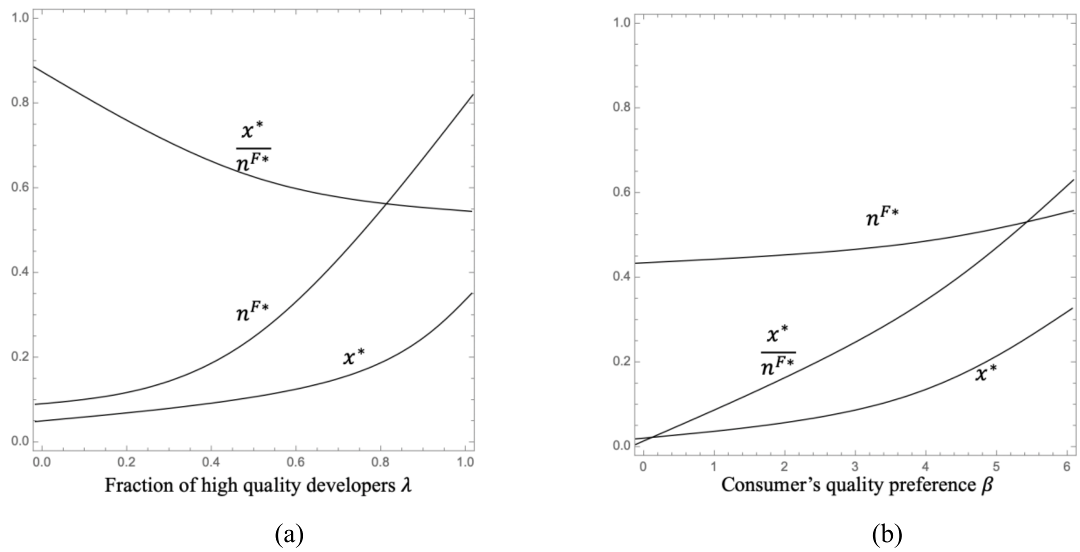



**Figure 2.** The number of developers, $n^{F*}$, first-party applications, $x^*$, and their ratio, $x^*/n^{F*}$ as functions of the fraction of high-quality developers, $\lambda$ (shown in (a)), and as functions of consumer's quality preference, $\beta$ (shown in (b)). $q_h$=0.4, $\theta_c = \theta_d$=2, $\alpha_c = \alpha_{dh}$=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).

Proposition 3.2 states that the platform will also only offer first-party applications when the fraction of high-quality developers is smaller than a threshold. In addition, if the consumer network effect $\alpha_c$, the consumer quality preference $\beta$, or the value of the high quality $q_h$ is too low, or the platform operating cost $\eta$ is too high, first-party application is not a preferred strategy. Under these scenarios, the added value by providing more high-quality applications is low because the increase in network effect (through both application variety and quality) is not strong enough to offset the platform maintenance cost and development cost. Otherwise, investments in developing first-party applications will lead to higher platform profit.

Part (2) suggests that when the market is fraught with low-quality applications, the platform is also more likely to offer a vertically integrated platform with no outside participation. This is because offering first-party applications will lead to great externality as a large fraction of low-quality developer freeride the quality improvement. Therefore, the platform raises the 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. Therefore, for the plight that the video game console maker, Atari, found itself in during the 1980s, our analyses suggest that closing the platform to outside participation might have been a sensible decision.

Intuitively, when the fraction of low-quality developers, $1-\lambda$, increases, one would expect that the platform offers 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 application, the platform is unable to completely internalize the development cost because it lacks the ability to price discriminate and has to set a uniform access fee to both high-quality and low-quality developers. Charging a high access fee to developers will discourage high-quality developers and therefore weakens the effectiveness of quality improvement, but charging a low access fee will allow more low-quality developers to enter and freeride the quality improvement (and the resulting larger consumer base) brought about by the first-party applications, which dilutes the effect of improving quality. 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 $\lambda$, the equilibrium number of third-party applications $n^{F*}$ increases at a much faster rate. This is because as $\lambda$ increases, application quality improvement comes from both increase in the number of high-quality third-party developers (a first order effect) and more first-party applications offered by the platform (a second order effect through $x^*$), leading to much higher incentives for developers to join. As a result, with a large $\lambda$, the platform is more likely to be dominated by third-party applications.

Part (4) indicates that when consumers have higher quality preference, not surprisingly, the platform is willing to develop more first-party applications. With a high $\beta$, the platform is better able to recover a large part of the development costs from the consumer side, taking advantage of the consumers' willingness to pay for quality. While the number of third-party developers $n^{F*}$ also increases with consumer quality preference, 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). As a result, with a high value of $\beta$ the platform is more likely to be dominated by first-party applications.

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

***Proposition 3.3: Comparison between First-party Application and Benchmark ($\mu$=0)***

*When consumer quality preference depends only on total quality, i.e., $\mu = 0$, first-party application 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 first-party application is also an effective quality regulation strategy for the platform. The benchmark model without quality regulation can be viewed a special case of first-party application with $x = 0$. Therefore, it is not surprising that the platform will fare better under first-party application. As average quality improves, the platform attracts more consumers to join, which makes the platform more appealing to developers through indirect network effects. However, the introduction of first-party applications also intensifies competition, 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 even exclude outside developer participation altogether. We also show that the platform is able to charge higher access fees to consumers to increase revenues. The increased revenues from access fees would be sufficient to offset the development cost for the 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 the comparisons under $\mu > 0$ are intractable, we start with the case where $\mu = 0$, and then consider the generalizability of the findings under $\mu > 0$.

### 5.1 Optimal Quality Regulation Strategy under $\mu = 0$

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

#### 5.1.1 Exclusion vs. Subsidization

In the following proposition, we characterize the choice between exclusion and subsidization from the platform's perspective.

***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 does, $\bar{q}^{E*} \geq \bar{q}^{S*}$;*

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

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 $\gamma$ accordingly to offset the high access fee to attract the desired amount of high-quality developers to join, achieving the same effect as exclusion. Therefore, subsidization is a more general and flexible quality regulation strategy as compared to exclusion.

However, as indicated by part (2) of proposition 4, an advantage of exclusion is that it does achieve higher average quality than subsidization (and higher than first-party application as well). Recall that the average quality under exclusion is $\bar{q}^{E*} = q_h$, which is the highest average quality level can be possibly achieved by the platform in our model setting. However, the highest level of quality is not always desirable to a platform, which explains why subsidization dominates exclusion: with subsidization, the platform can balance between application quantity and quality, while exclusion is more rigid 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 many others such as subsidization and first-party application. This might explain why exclusion is commonly used in practice although it is not necessarily the profit-optimizing strategy. It is worth noting that there might be other reasons for exclusion being a widely adopted strategy in practice. For example, in our model we focus on "goods" produced by outside developers rather than "bads" that may be harmful to consumers, whose presence on the platform may cause disutility.

Except for average quality, subsidization dominates almost every other front 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 higher revenue that is enough to offset the cost of subsidization to earn 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 application, which will be discussed 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 Application***

*The platform's optimal choice of quality regulation strategy between subsidization and first-party content can be characterized as:*

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

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

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

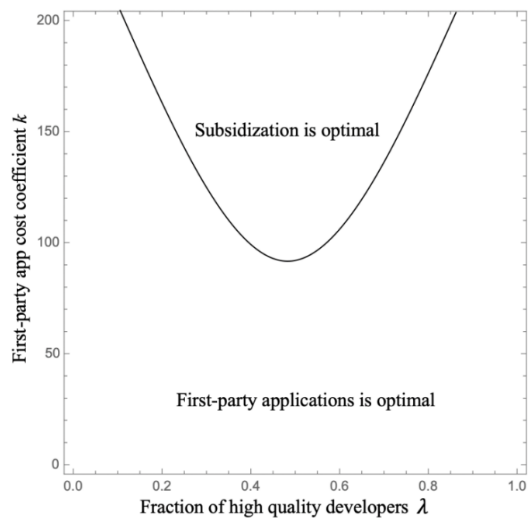



**Figure 3.** The platform's optimal quality regulation strategy. $\beta$=2.7 , $q_h$ =0.4, $\theta_c = \theta_d$=2, $\alpha_c = \alpha_{dh}$=0.6, $\alpha_{dl}$=0.5, $w$=1, $b$=1.5, and $\eta$=0.6.

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

Figure 3 illustrates the platform's optimal choice between subsidization and first-party application graphically on the plane of the first-party development cost, $k$ and the fraction of high-quality developers, $\lambda$.

The following proposition provides further insight into the optimal quality regulation strategy for the platform.

***Proposition 6: Network Sizes and Quality Level under Optimal Strategy***

*When the platform chooses between subsidization and first-party application, 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 application) that can enable 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 to improve its profitability. In Figure 4, we illustrate the platform's optimal choice of quality regulation strategy when its objective is to maximize average application quality and contrast this choice to the profit maximizing choice described in proposition 6.

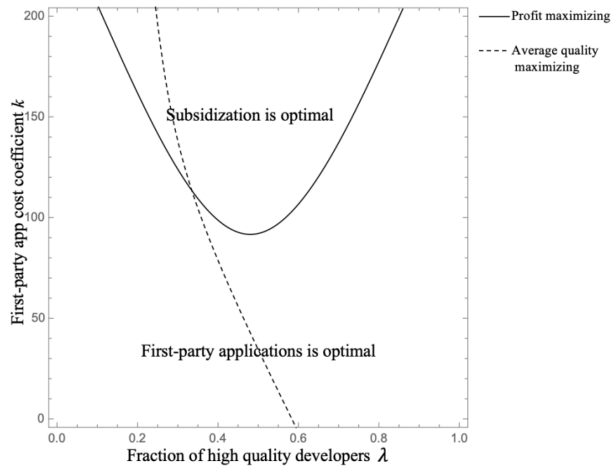



**Figure 4.** Profit maximizing strategy vs. average quality maximizing strategy. $\beta=2.7$, $q_h=0.4$, $\theta_c=\theta_d=2$, $\alpha_c=\alpha_{dh}=0.6$, $\alpha_{dl}=0.5$, $w=1$, $b=1.5$, and $\eta=0.6$.

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

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

Figure 5 illustrates how $\mu$ affects the platform's optimal choice between subsidization and first-party application. The curve for $\mu = 0$ in Figure 5 corresponds to the curve we have shown in Figure 3. The two curves for $\mu = 1$ and $\mu = 2$ allow us to see how $\mu$ potentially impacts the platform's optimal choice.

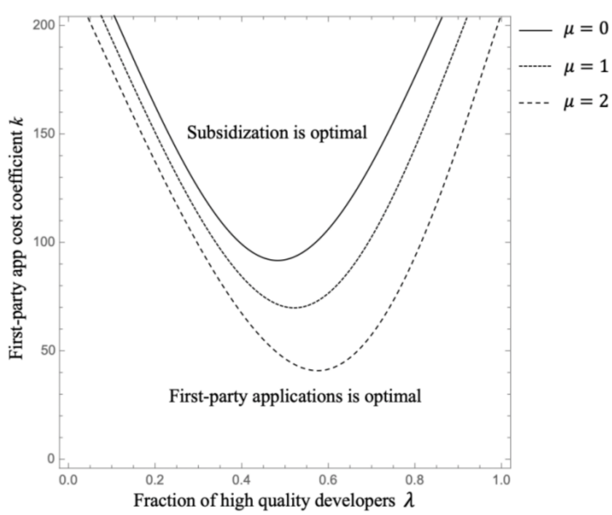


**Figure 5.** The platform's optimal quality regulation strategy under different values of $\mu$. $\beta$=2.7, $q_h$=0.4, $\theta_c = \theta_d$=2, $\alpha_c$=0.6, $\alpha_{dh}$=0.6, $\alpha_{dl}$=0.5, $w$=1, $b$=1.5, and $\eta$=0.6.

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

## 6. Social Welfare Analyses

We have studied how different quality regulation strategies can be employed to improve the 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 where $\mu = 0$, 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 $t \in \{0, E, S, F\}$, the social welfare is the sum of total consumer utility $V^*$, total developer utility $U^*$, and the platform's profit $\Pi^{t*}$, defined as

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

The equilibrium social welfare under different quality strategies -- benchmark $W^{0*}$, exclusion $W^{E*}$, subsidization $W^{S*}$, and first-party application $W^{F*}$ -- are characterized in Appendix 1. The properties of social welfare under the different strategies are summarized in the following proposition.

***Proposition 7: Social Welfare under Optimal Strategy***

(1) *Subsidization always improves both the platform's profit and the social welfare, i.e., $\Pi^{S*} \geq \Pi^{0*}$ and $W^{S*} \geq W^{0*}$. While first-party application always improves the platform's profit, i.e., $\Pi^{F*} \geq \Pi^{0*}$, it does not necessarily improve the 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 application does, $W^{S*} > W^{F*}$. However, the opposite is not necessarily true. Therefore, a social planner would choose subsidization over first-party application more often than the platform would.*

We have discussed in the previous section that exclusion does not necessarily improve the platform's profit, while both subsidization and first-party application do. According to part (1) of proposition 7, subsidization also surely improves the social welfare, because the subsidization entails an internal transfer between the platform and high-quality developers. In contrast, first-party application might not always increase social welfare, because the 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 the social welfare. Recall that exclusion reduces the network sizes, especially the developer network size, increases the access fee to consumers, and could lower the profit for the platform, which are all detrimental to social welfare.

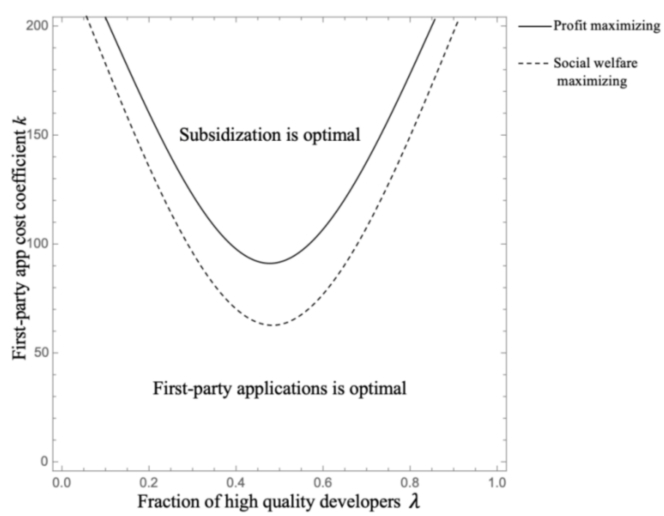



**Figure 6.** Profit-maximizing quality regulation strategy vs. social welfare-maximizing quality regulation strategy. $\beta$=2.7, $q_h$=0.4, $\theta_c = \theta_d$=2, $\alpha_c$=$\alpha_{dh}$=0.6, $\alpha_{dl}$=0.5, $w$=1, $b$=1.5, and $\eta$=0.6.

Part (2) of the proposition suggests that subsidization, in addition to being profit improving, is also the most social welfare friendly quality regulation strategy among the three. In fact, when subsidization is optimal, or profit-maximizing for the platform, it is always social welfare-maximizing. However, when first-party application is optimal for the platform, it may not be social welfare-maximizing, which suggests that a welfare-maximizing social planner prefers 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. As we can see, the area under which subsidization is optimal is larger for the social planner and subsumes that for the platform, implying that the social planner would be more likely to choose subsidization as the optimal quality regulation strategy than the platform would.

### 6.2 Social Welfare under $\mu > 0$

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

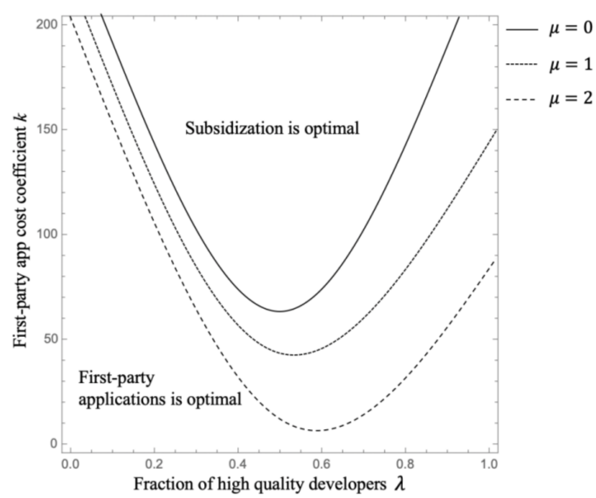



**Figure 7.** Social welfare-maximizing quality regulation strategy under different values of $\mu$. $\beta$=2.7, $q_h$=0.4, $\theta_c$= $\theta_d$=2, $\alpha_c$=0.6, $\alpha_{dh}$=0.6, $\alpha_{dl}$=0.5, $w$=1, $b$=1.5, and $\eta$=0.6.

## 7. Extensions

We consider two extensions of the model to investigate how changes in some model parameters and assumptions may impact the findings we obtained in previous sections. Specifically, we study the case where 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 where $\mu = 0$ throughout this section. The equilibrium outcomes of these extensions, along with the assumptions to ensure the platform's problems to be well-defined, are presented 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 that development cost is increasing in 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 $C(q)\theta_i$, with

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

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 given by:

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

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 the above 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 where high-quality developers incur a development cost* $c\theta_i$ *where* $c > 1$*:*

*(1) Under subsidization, the optimal subsidy is decreasing in c. Under first-party applications, the optimal*

*number of first-party applications is not monotone 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 all models, 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 strategy of exclusion, because $c$ is merely a constant scalar on high-quality developers' development cost. However, due to high-quality developers' cost disadvantage, they are less likely to join the platform in comparison to the benchmark model with everything else being equal. As $c$ increases, fewer high-quality developers would join the platform in the absence of platform intervention. Although providing a higher amount of 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 price to low-quality developers at the same time when it subsidizes high-quality developers). As a result of the tradeoff between quantity and quality, the platform reduces the amount of subsidy offered to high quality developers when $c$ is higher. Most structural results under subsidization we presented in Section 4.2 carry over. Under first-party application, 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 freeriding the quality improvement. As a result, the platform raises the access price high enough to exclude outside participation altogether.

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

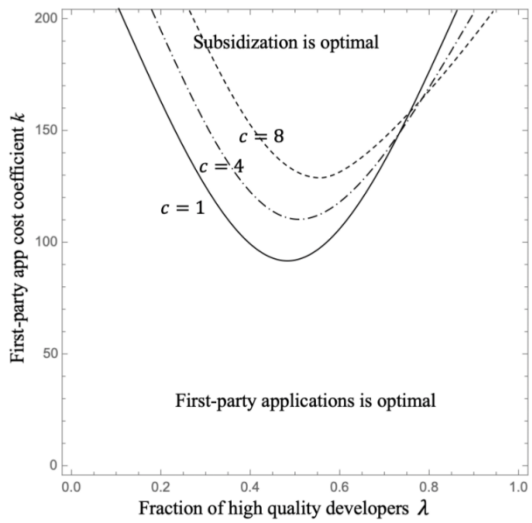



**Figure 8.** The platform's optimal quality regulation strategy when $c=1$, $c=4$ and $c=8$. $\beta=2.7$, $q_h=0.4$, $\theta_c=\theta_d=2$, $\alpha_c=0.6$, $\alpha_{dh}=0.6$, $\alpha_{dl}=0.5$, $w=1$, $b=1.5$, and $\eta=0.6$.

### 7.2. Concave First-party Application Development Cost

We have assumed a convex development cost for first-party applications in section 4.3. However, application development may exhibit economies of scale as 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 and Kemerer 1989). We consider a concave first-party application development cost function in which the development cost takes the form of $k\sqrt{x}$ for developing $x$ first-party applications. The following proposition compares the properties of the optimal number of first-party application 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 is decreasing in the fraction of high-quality developers* $\lambda$*, whereas it is increasing in* $\lambda$ *when the development cost is* $kx^2$*.*

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

When the fraction of high-quality developers $\lambda$ 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 the development cost as less low-quality developers would freeride, which increase the return on developing more first-party applications. On the other hand, a larger fraction of high-quality developers also puts a limit on the quality improvement that can be achieved through developing first-party applications. When the development cost is $kx^2$, as we discussed proposition 3.2, the former factor dominants the latter because, with a convex cost, the development cost goes up quickly with more first-party application. The optimal number of first-party applications $x^*$ would therefore increase as $\lambda$ 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 as it goes up much slower as more first-party applications are developed. Therefore, the latter effect of increasing $\lambda$ dominates the former. When $\lambda$ is low, the platform can afford to develop a large amount of first-party applications to improve the quality without incurring too high a cost due to economies of scale. As $\lambda$ increases, the marginal quality improvement associated with first-party applications becomes weaker. So, the platform would gradually pull back on the amount of the first-party applications. Part (2) of the proposition suggests that with economies of scale, as expected, the concave cost function $k\sqrt{x}$ does lead to higher optimal number of first-party applications as compared to the convex cost function $kx^2$.

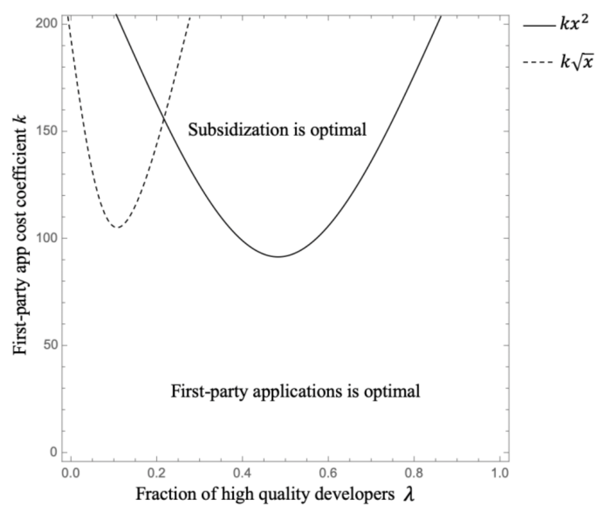



**Figure 9.** The platform's optimal quality regulation strategy under different first-party development costs. $\beta=2.7$, $q_h=0.4$, $\theta_c=\theta_d=2$, $\alpha_c=0.6$, $\alpha_{dh}=0.6$, $\alpha_{dl}=0.5$, $w=1$, $b=1.5$, and $\eta=0.6$.

Changing the cost function from convex to concave does not strictly makes either subsidization or first-party application strategy more attractive. Figure 9 provide an example to illustrate the shift in the choice of strategies. As it shows, with cost $k\sqrt{x}$, subsidization becomes more attractive relative to first-party application when $\lambda$ is small, whereas first-party application becomes more favorable when $\lambda$ is large.

## 8. Discussion and Conclusions

With platforms becoming an increasingly popular business model in the technology industry, the role of a platform company transits from coordinating internal economic activities to one that also includes providing boundary resources to outside complementors as well as regulating the conduct of firms within its platform ecosystem (Boudreau and Hagiu 2009). While prior literature has provided many insights into the pricing strategies in a two-sided market (Hagiu 2006, 2009b, Jeon and Rochet 2010), the regulation of the quality of complementary applications has so far received little research attention, which forms the central research question in this study. We compare three strategies that are widely employed in practice: excluding access to low-quality complementors, providing a subsidy to high-quality ones, 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 profit and will often result in greater social welfare as well. Interestingly, strategies aimed at increasing application variety and those aimed at improving application quality need not be in conflict with one another as suggested by prior research (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 its unique advantages and limitations. Under exclusion, a platform can achieve a high quality level with a 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 does improve both due to its power of price discrimination, and is a particularly attractive choice if the platform faces a high first-party development cost. 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, first-party application strategy works particularly well if platform development cost is low, but such a strategy may suffer from over-provision of first-party applications to the extent that it hurts social welfare and may lead to a vertically integrated platform that excludes outside participation. In addition, under first-party application the platform faces challenge in internalizing development cost due to the freeriding of low-quality developers, and the issue is most prominent when quality provision by third-party is more evenly distributed.

**Table 2. Summary of Findings (under $\mu = 0$)**

| | **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 profit and smaller networks | Becomes inefficient when $\lambda$ is either too big or too small | Causes freeriding by low-quality developers; may lead to a closed platform |
| As compared to benchmark (without quality regulation) | Not necessarily improve profit or network sizes | Improve profit and network sizes of both developers and consumers | Improve profit and consumer network size |
| Equilibrium average quality | Highest | Higher than benchmark | Higher than benchmark |
| When is the policy optimal | Never, dominated by subsidization | When $k$ is high and the distribution of high- vs. low-quality developers is more even | When $k$ is low; or when $k$ is high and the distribution of high- vs. low-quality developers is more lopsided |
| Implication for social welfare | Not always social welfare improving | Always social welfare improving | Not always social welfare improving |

Our research also reveals a number of important managerial implications for practitioners. For example, although the strategy of exclusion appears intuitively appealing, it may lead to unintended consequences under certain contexts, and therefore its adoption should be carefully weighed against other alternatives. In contrast, platform designs that involve subsidizing high-quality complementors, like 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, with many platforms – such as Netflix – start integrating into content provision and investing aggressively in the development of their exclusive first-party applications, managers need to carefully evaluate whether choosing such a strategy is advantageous, taking into consideration factors such as their cost efficiency in relative to outside developers, and the quality distribution among third-party applications. Our study here provides some 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, such as directing more transactions to them through a recommender system, the need for subsidizing high-quality developers is greatly reduced.

A number of limitations of our model points to several directions for future research. First, while we assume third-party quality provision is exogenously determined, another useful quality regulation strategy can aim at incentivizing low-quality producers to exert effort and improve the quality of their applications, which will endogenize the application development stage. Second, one of the reasons that many platforms deny access to some third-party developers is to exclude harmful applications, whose presence in the platform leads to negative network effects. By considering both economic goods and bads in the same model, one can potentially derive 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 application at the same time, may provide further insights on how limitations of employing 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., C.F. Kemerer. 1989. Scale Economies in New Software Development. *IEEE Transactions on Software Engineering*. **15**(10) 1199-1205.

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

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

Boudreau, K. 2010. Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. *Management Science*. **56**(10) 1849-1872.

Boudreau, K.J., A. Hagiu. 2009. Platform rules: multi-sided platforms as regulators. A. Gawer, ed.

*Platforms, Markets and Innovation*, Edward Elgar Publishing, 163-191.

Casadesus-Masanell, R., G. Llanes. 2015. Investment Incentives in Open-Source and Proprietary Two-Sided Platforms. *Journal of Economics and Management Strategy*. **24**(2) 306-324.

Ceccagnoli, M., C. Forman, P. Huang, D. Wu. 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., E. Katsamakas. 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., G. Parker, M.W. Van Alstyne. 2006. Strategies for two-sided markets. *Harvard Business Review*. **84**(10) 92-101.

Eisenmann, T.R., G. Parker, M. Van Alstyne. 2009. Opening platforms: how, when and why? A. Gawer, ed. *Platforms, markets and innovation*, Edward Elgar Publishing, 131-162.

Gawer, A., M.A. Cusumano. 2002. *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. Harvard Business School Press, Boston, MA.

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

Gawer, A., R. Henderson. 2007. Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel. *Journal of Economics & Management Strategy*. **16**(1) 1-34.

Ghazawneh, A., O. Henfridsson. 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, Harvard Business School Strategy Unit Working Paper.

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

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

Jeon, D.-S., J.-C. Rochet. 2010. The Pricing of Academic Journals: A Two-Sided Market Perspective. *American Economic Journal: Microeconomics*. **2**(2) 222-255.

Kim, J.-H., J. Prince, C. Qiu. 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., S. Li, A.B. Whinston. 2011. Innovation and price competition in a two-sided market. *Journal of*

*Management Information Systems*. **28**(2) 171-202.

Parker, G., M.V. Alstyne. 2017. Innovation, Openness, and Platform Control. *Management Science*. **64**(7) 3015-3032.

Parker, G.G., M.W. Van Alstyne. 2005. Two-Sided Network Effects: A Theory of Information Product Design. *Management Science*. **51**(10) 1494-1504.

Parker, G.G., M.W. Van Alstyne, S.P. Choudary. 2016. *Platform Revolution: How Networked Markets Are Transforming the Economyand How to Make Them Work for You*. WW Norton & Company.

Rochet, J.C., J. Tirole. 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., L. Xue, A. Rai, C. Zhang. 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., B. Konsynski, A.A. Bush. 2010. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*. **21**(4) 675-687.

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