We examine how exercising control over a technology platform can increase profits and innovation. By choosing how much to open and when to bundle enhancements, platform sponsors can influence choices of ecosystem partners. Platform openness invites developer participation but sacrifices direct sales. Bundling enhancements early drives developers away but bundling late delays platform growth. Ironically, developers can prefer sponsored platforms to unmanaged open standards despite giving up their applications. Results can inform innovation strategy, antitrust and intellectual property law, and the management of competition.

Keywords: Sequential Innovation, Platforms, Copyright Length, IT Systems Design, Network Effects, Network Externalities, Bundling, Openness.

Acknowledgments: This work has benefited from comments and suggestions by Marcel Canoy, Jovan Grahovic, Josh Lerner, Sadao Nagaoka, Marc Rysman, Patrick Waelbroeck and Thomas Weber. Nico Savva provided valuable guidance for our model of technological uncertainty. Seminar participants from Cambridge University, IDEI / Bruegel, Imperial College, Kansai University, Law and Economics Consulting Group, MIT, Stanford Institute for Economic Policy Research, The University of California-San Diego, and The University of Washington also helped shape this research. The National Science Foundation provided support via grants IIS-0338662 and SES-0323227. Funding from Cisco Systems Inc. and Microsoft Corp. is also gratefully acknowledged.
1 Introduction

Platform business models have become a ubiquitous feature of the information economy. Common products such as personal computers, cell phones, gaming systems, streaming media, and telecommunications infrastructure can be described in terms of systems where developers build “applications” on top of a “platform.” When there is sequential innovation such that access to early developer innovations stimulates later developer production, platforms that enforce developer disclosure can grow more quickly than those that do not. However, due to a prisoners’ dilemma that arises in the disclosure game, developers share less than is socially optimal in the absence of a contract enforced by the platform sponsor. Hence platform-system growth depends critically on the control a platform exercises over developer disclosure as a condition for participating in the platform ecosystem.

To build the ecosystem, platform sponsors often embrace modular technologies and encourage partners to supply downstream complements (Baldwin & Clark, 2000; Baldwin & Woodard 2008; Boudreau, 2008). Loose integration promotes layered industries. In the personal computer industry, for example, these layers consist of semiconductor manufacture, PC assembly, operating system provision, and application software, among others (Grove, 1996; Baldwin & Clark, 2000). The credit card and telecommunications industries are similarly layered (Evans, et. al., 2006).

As a result of the increasing economic importance of platform ecosystems, a growing literature has focused on platform design (e.g., Cusumano & Gawer 2002), platform economics, and the associated business strategies for managing them (see, e.g., Boudreau, 2007; Bresnahan & Greenstein, 1999; Farrell, Monroe, & Saloner, 1998). Recent literature conceives of platforms as mediating markets with two-sided network externalities and analyzes pricing across potentially distinct user groups (Caillaud & Jullien, 2003; Parker & Van Alstyne, 2000, 2005; Rochet & Tirole, 2003). Choosing the optimal level of openness is also critical
for firms that create and maintain platforms (Gawer & Cusumano, 2002; West, 2003; Parker & Van Alstyne, 2005; Gawer & Henderson, 2007; Boudreau, 2008; Eisenmann, 2008). This decision entails a tradeoff between growth and appropriability (West, 2003). Opening a platform can spur growth by harnessing network effects, reducing users lock-in concerns, and stimulating downstream production. At the same time, opening a platform typically reduces users switching costs and increases competition, reducing sponsors’ ability to capture rents.

To date, however, there has been little formal modeling to address the question of how a platform sponsor can exercise non-price control in order to capture profits and promote growth in the platform ecosystem (Boudreau & Hagiu, 2009). Earlier non-price work has focused at the regulatory level to explore the question of how to promote initial innovation by allowing an innovator to capture profits from follow-on innovations. This literature has looked primarily at patent length and breadth, leaving open the question of how firms themselves could use these results (e.g., Chang, 1995; Gilbert & Shapiro, 1990; Green & Scotchmer, 1995; Klemperer, 1990; Landes & Posner, 2003).

We build on the sequential innovation literature and broaden the analysis to the following set of questions: When should a platform sponsor open a resource to outside development? How does competition affect openness? How does the ability to reuse platform assets affect the level of openness? Does the number of downstream developers or their added value affect openness? If downstream developers do add value, should the firm privately subcontract with a subset or should the firm open the platform to the entire developer pool? When should a platform fold new developer applications into the platform?

To address these questions, we develop a tractable model of downstream production by developers who add value to a platform by producing applications. We conceive of a platform as the components used in common across a product family (Boudreau, 2007) whose functionality can be extended by applications and is subject to network effects (Parker & Van Alstyne, 2005; Eisenmann et. al. 2006; Evans et. al 2006). A platform is “open”
to the extent that it places no restrictions on participation, development, or use across its distinct roles, whether developer or end-user (Eisenmann, et. al. 2009). Further openness, unrestricted participation at the sponsor level, we analyze as a fully unrestricted open standard. We focus on the set of platforms that grow primarily through sequential innovation. Our model incorporates the ability to reuse output from one period as production input in the next period. Developers can then incorporate platform assets into their applications development. Further, the development of second generation applications can depend on the value and quantity of applications developed in the first. The tradeoff is that converting assets from closed to open sacrifices salable assets, thus creating a tension between current and future profits. The model allows us to demonstrate (i) the optimal level of openness for a platform (ii) when downstream applications should also become open in order to promote second generation application production, (iii) when a platform sponsor should use closed subcontracts instead of decentralized open innovation, (iv) how competition affects openness, and (v) why the presence of a platform sponsor that forces openness on downstream developers can make even developers themselves (as well as users) better off. The regulatory implication is that sponsors need longer term property rights than developers in order to effectively manage downstream innovation. Section 2 develops the model and main results, including social welfare, competition, and technological uncertainty. Section 3 considers alternate organizational forms. We consider extensions in 4 and conclude in Section 5.

2 The Model

Consider a model of platform innovation with two periods of equal length $t$ and discount rate $r$. The market ecosystem includes platform sponsors, developers, and consumers. We consider three points of value. The first, denoted as $V$, is the value of the platform independent of developer applications and add-ons. The second and third are the value created by
developer production in periods 1 and 2, denoted as $y_1$ and $y_2$. At the start of period 1, a platform sponsor makes fraction $\sigma$ of a platform of value $V$ openly available to developers so that developers can produce for the platform.\(^1\) Hence the first period open code base is $\Omega_1 = \sigma V$. At the beginning of each period $i$, $i = 1, 2$, developers invest a fixed cost $F$ and variable cost $c y_i^2$ to produce output $y_i$ that has a per-unit value of $v$ to consumers. Developers produce according to a standard Cobb-Douglas production function where $k$ is a reuse coefficient that determines the level of conversion between a stock of code resource, $\Omega$, into applications and add-ons. A technology parameter, $\alpha$, determines the efficiency of production so $y_i = k \Omega_i^\alpha$. Platforms choose $\sigma$ and $t$. The remaining terms, $V, v, k, c, F, \alpha$, and $r$, are exogenous. We discuss potential relaxations to these assumptions in section 4.

To build intuition and develop necessary building blocks, we first analyze the model without competition, developer choice, or costs. After determining platform choices in isolation, we consider social planner choices. We then explore the effects of technological change and the number of developers. We analyze competition at both the platform and developer levels. We conclude by analyzing the participation game that developers face in the absence of strong contract enforcement. Our solution concept in the developer game is subgame perfect Nash equilibrium. Modeling $t$ as a choice variable is non-standard and requires explanation. Current U.S. copyright laws provide exclusive protection for 95 years for corporate authorship. European laws provide similarly long protection. A core finding of our model is that one-size-fits-all copyright duration at both the platform and application level is neither socially optimal, nor profit-maximizing in the context of sequential innovation. Hence we focus on directly analyzing the duration of protection for follow-on innovation.

We follow Chang (1995) in assuming that consumers share a common value $V$ for the platform and $v$ for each unit $y$ of application produced. We assume that leakage to consumers

\(^{1}\)Openly exposing a platform’s Application Programming Interfaces (APIs) is a common way to provide a small fraction of code to enable developers to produce for the platform.
results in a net loss of platform profit in the amount of the first period giveaway, $\sigma V$. Technological obsolescence prevents developers from reusing open resources more than once (further reuse would increase the value of openness). Thus, second period open stock, which developers use as a production input, is the period 1 production s.t. $\Omega_2 = y_1 = k(\sigma V)^\alpha$. Developer output in periods 1 and 2 can be expressed as $y_1 = k(\sigma V)^\alpha$ and $y_2 = k^{1+\alpha}(\sigma V)^\alpha^2$. Section 2.2 considers a direct licensing contract to avoid the loss of platform value. To make revenue streams comparable, second period revenue is discounted to the end of period one at rate $r$.

Let $t$ be the length of the exclusionary period offered to developers during which they can sell their applications at positive profits. That is, analogous to a period of patent protection, $t$ represents the time before which a sponsor agrees not to compete with the developer, but after which the sponsor will fold new developer features into the open platform. Newly open features from one developer then become available to all. To facilitate analysis, we combine parameters $r$ and $t$ into discount coefficient $\delta = e^{-rt}$. Time is bounded by $0 \leq t < \infty$ which restricts $\delta$ to the range $0 < \delta \leq 1$. Price is then determined by the length of time before an application is forced into the open domain. Consumers know that applications will be freely available after the exclusionary period $t$. Therefore, developers can charge consumers only for the difference between the full value of the product today and the discounted value of the product when it becomes open and free. Thus, $p = v - \delta v = v(1 - \delta)$. If the platform never bundled new developer applications into the platform ($t \to \infty$) then $\delta \to 0$ and $p = v$. Likewise, if the exclusionary period ends immediately ($t = 0$), then $\delta = 1$ and $p = 0$.

We make the same assumption as in Green and Scotchmer (1995) that Nash bargaining governs the revenue split on downstream innovation, giving each party $\frac{1}{2}$ the downstream developer-produced surplus.\(^2\) We assume zero marginal production costs and a sufficiently

\(^2\)In practice, licensing encourages growth through openness but “indexes the sponsor’s share of profits to platform expansion in a low friction way.” (Interview Source: Guido Jouret, CTO Emerging Markets Group, Cisco Systems Inc. 9-8-2006). A Nash bargain is thus a reasonable approximation.
large value added, $V$ and $v$, to cover platform and developer fixed costs. For many information goods and even physical goods such as semiconductors, zero marginal cost is a reasonable approximation. Regardless, we consider costs in section 2.2. Developer profit and platform sponsor profits can then be written as

$$\pi_d = \frac{1}{2}py_1 + \delta \frac{1}{2}py_2$$  \hspace{1cm} (1)$$

$$\pi_p = V(1 - \sigma) + \frac{1}{2}py_1 + \delta \frac{1}{2}py_2$$  \hspace{1cm} (2)$$

Expressing platform sponsor profit in terms of model primitives yields

$$\pi_p = V(1 - \sigma) + \frac{1}{2}v(1 - \delta)k(\sigma V)^{\alpha} + \delta \frac{1}{2}v(1 - \delta)k^{1+\alpha}(\sigma V)^{\alpha^2}. \hspace{1cm} (3)$$

### 2.1 Platform Sponsor Choice of $\sigma$ and $\delta$

We now explore the central tension facing the platform sponsor: the degree to which it should sacrifice direct platform profits in order to stimulate downstream innovation, and its commitment to avoid competing directly with developers before expiration of the proprietary period. The optimal contract is a pair $\langle \sigma, \delta \rangle$ (isomorphic to $\langle \sigma, t \rangle$) where choice parameters $\sigma$ and $\delta$ represent the share of value (level of openness) used to subsidize developers, and the period of proprietary developer protection. The amount of production in each period, $y_1$ and $y_2$, the discount rate $r$, and the responsiveness of production to openness will govern a platform sponsor’s choices. We assume a convex region of interest, defined by a negative semidefinite matrix with respect to openness and time. Thus it must satisfy the standard Hessian conditions for a two dimensional optimum. We develop conditions for optimal openness in terms of elasticities. The elasticity of output in each period with respect to $\sigma$ is

$$\eta_i = \frac{\partial y_i}{\partial \sigma} \frac{\sigma}{y_i}, \ i = 1, 2.$$
Proposition 1  The platform sponsor’s optimal choice $\sigma^*$ is defined by the ratio of production revenues gained to subsidy revenues lost weighted by the elasticity of output per period.

$$\frac{(1/2) p y_1}{\sigma V} \eta_1 + \frac{(1/2) \delta p y_2}{\sigma V} \eta_2 = 1$$  \hspace{1cm} (4)

Proof. To establish the result, first calculate the first-order condition on platform profit with respect to $\sigma$:

$$\frac{\partial \pi_p}{\partial \sigma} = -V + \alpha \frac{1}{2} p k \sigma^{a-1} V^\alpha + \alpha^2 \frac{1}{2} \delta p k^{1+a} \sigma^{a-1} V^\alpha = 0.$$  \hspace{1cm} (5)

Add $V$ to both sides and premultiply all terms by $\sigma$ to restore the expressions for output $y_1$ and $y_2$. Divide through by $\sigma V$. Cobb-Douglas models yield, $\eta_1 = \alpha$ and $\eta_2 = \alpha^2$. Substituting $\eta$ terms for $\alpha$ terms provides the required result. \hfill \blacksquare

Intuitively, when the platform sponsor opens its core resources to outside parties, the gain from sharing in developer profits must offset platform losses (forgone revenue $\sigma V$). Opening the platform and subsidizing developers stimulates downstream production. The responsiveness of output to openness is captured in the elasticity terms so that the optimal level of openness properly balances the revenues lost and gained.

We now explore the platform sponsor’s choice of time during which developers enjoy proprietary protection for their innovations.

Proposition 2  The optimal choice $\delta^*$ is governed by the ratio of developer production in periods 1 and 2.

$$\delta^* = \frac{1}{2} \left(1 - \frac{y_1}{y_2}\right)$$  \hspace{1cm} (6)
This implies that the condition for a finite duration to protection of platform applications is higher second period output. A further implication is that it is never profit maximizing to force the immediate free release of developer applications.

**Proof.** To establish the result for $\delta$, calculate the first-order condition on platform profit with respect to $\delta$. Since $\delta$ terms do not appear in output, we express profit in terms of $y_1$ and $y_2$.

$$\frac{\partial \pi_p}{\partial \delta} = -y_1v + y_2v(1 - \delta) - \delta y_2v = 0,$$

(7)

Rearranging terms provides the required result. ■

Expressing the requirement for finite copyright duration, $y_2/y_1 > 1$ in terms of model primitives produces $\frac{k^{1+\alpha} \sigma^2 V^{\alpha}}{\sigma \alpha V^{\alpha}} > 1$. Raising both sides to $1/\alpha$, this reduces to $\frac{k^{1/\alpha}}{\sigma^{1-\alpha} V^{1-\alpha}} > 1$. Clearly, a larger reuse coefficient, $k$, makes the existence of a second period more likely, while a larger platform value $V$ makes a finite time less likely. A further re-arrangement of this condition is that $y_1 > \sigma V$ which implies that there is never a second period unless the first period developer production exceeds the amount of the platform that is given away.

In Corollary 1 below, we explore the effect of model primitives on the platform sponsor’s choice variables. Time moves in the opposite direction from the discount coefficient $\delta$. We provide detailed derivations in the Appendix.

**Corollary 1** Comparative Statics – The following table summarizes effects of model primitives on platform sponsor choices of optimal contract.

<table>
<thead>
<tr>
<th></th>
<th>$\sigma^*$</th>
<th>$\delta^*$</th>
<th>$t^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform value: $V$</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Developer value: $v$</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Reuse coefficient: $k$</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Technology: $\alpha$</td>
<td>indeterminate</td>
<td>indeterminate</td>
<td>indeterminate</td>
</tr>
</tbody>
</table>
**Proof.** To produce sensitivity analyses, apply the envelope theorem to determine the sign of each effect. Derivations appear in the Appendix.

Rising platform value $V$ implies closing the platform more and folding in new features later. Equation 4 shows this directly for $\sigma^*$ since $V$ only appears as part of $\sigma V$. A more valuable initial platform means that less of its value can be sacrificed to stimulate developer production. Interestingly, rising platform value also lengthens $t^*$ the proprietary period offered to developers. In effect, larger $V$ implies the sponsor prefers to take profits directly instead of relying on indirect downstream innovation.

In contrast, increasing the developer value, $v$, per unit produced has the effect of increasing the sponsor’s willingness to open the platform. The sponsor rationally sacrifices direct platform profits in order to share in rising developer surplus. Somewhat surprisingly, an increase in the value of developer production leads a platform sponsor to offer developers a longer proprietary period $t^*$. Increased surplus in both periods has the effect of making the sponsor more patient. More valuable new features are folded into the platform later.

As the reuse coefficient, $k$, rises, developer production increases. This implies opening the platform more but, in contrast, implies folding new features into the platform sooner. The sponsor should sacrifice direct platform profits in order to stimulate indirect developer surplus. In this case, however, reuse is sufficiently important to subsequent second period innovation that the platform sponsor reduces the proprietary period in order to pull second round profits closer in time.

Finally, the technology coefficient, $\alpha$, has a non-monotonic effect on the platform sponsor’s optimal contract. Specific parameter values govern the choice of contract and can cause openness and time to both rise and fall.
2.2 Welfare

We extend the model to include developer fixed costs $F$ in each period and increasing marginal costs $c_y^{1/\alpha}$ where, for simplicity, marginal cost remains small enough that $vy_2 \geq \frac{c}{\alpha} y_2^{1/\alpha}$. We continue to assume a convex region of interest, defined by a negative semidefinite matrix with respect to openness and time. These additions allow us to compare the choices for a welfare optimum against those of a sponsor’s maximum net profit. Adding fixed and marginal costs to Equation 2 provides the basis for comparison.

$$\pi_p^c = (1 - \sigma)V + \frac{1}{2} \left( py_1 - cy_1^{1/\alpha} - F \right) + \frac{\delta}{2} \left( py_2 - cy_2^{1/\alpha} - F \right)$$

Including consumer surplus, the following welfare equation then determines the social planner’s optimization.

$$\arg \max_{\sigma, \delta} W = V + (vy_1 - cy_1^{1/\alpha} - F) + \delta (vy_2 - cy_2^{1/\alpha} - F)$$

Subject to a developer participation constraint:

$$\pi_d^c = \frac{1}{2} \left( py_1 - cy_1^{1/\alpha} - F \right) + \frac{\delta}{2} \left( py_2 - cy_2^{1/\alpha} - F \right) \geq 0.$$  (10)

A positive price, $p = v(1 - \delta) > 0$, represents a wealth transfer from consumers, while the platform subsidy $\sigma V$ represents a wealth transfer from the platform sponsor. Both are irrelevant to a social planner except to the degree that developers must cover development costs. Note that in the absence of costs, a social planner simply allocates all existing resources for innovation without delay and chooses $\langle \sigma_c^\dagger, t_c^\dagger \rangle = (1, 0)$.

---

This formulation includes the standard quadratic form $cy^2$ as a special case (i.e. $\alpha = \frac{1}{2}$) but allows cost to fall with improved technology.
**Proposition 3** The social optimum is a contract \( \langle \sigma_c^\dagger, \delta_c^\dagger \rangle \) with \( \delta_c^\dagger > \delta_c^* \) and \( \sigma_c^\dagger > \sigma_c^* \). The social planner prefers a more open platform and a shorter proprietary period \( (t_c^\dagger < t_c^*) \) for applications than do platform sponsors.

**Proof.** To establish the claim with respect to \( \delta \), solve the platform sponsor’s maximization problem inclusive of cost. Taking the first order condition of platform profit \( \pi_p \) w.r.t. \( \delta \) leads the platform sponsor to choose

\[
\delta_c^* = \frac{1}{2} \left( 1 - \frac{y_1}{y_2} - \frac{cy_1^{1/\alpha} + F}{vy_2} \right). 
\] (11)

The social planner chooses \( \delta \) subject to the participation constraint \( \pi_d^c \geq 0 \) for cost recovery. Solving for \( \delta \) produces two roots. Eliminate the negative root by choosing \( c = F = 0 \). In the absence of cost, the positive root reduces to \( \delta = 1 \). Hence, absent the need to recover cost, a social planner prefers to release developer additions immediately. Otherwise, the social planner chooses.

\[
\delta_c^\dagger = \frac{1}{2} \left( 1 - \frac{y_1}{y_2} - \frac{cy_1^{1/\alpha} + F}{vy_2} + \Delta \right). 
\] (12)

All terms except \( \Delta = \sqrt{4vy_2(y_1-cy_2^{1/\alpha}+F) + ((vy_2-cy_2^{1/\alpha}+F)-y_1)^2} \) are the same as those chosen by the platform sponsor. Observing that \( \Delta \) is the positive root completes the claim. Also note that \( \delta_c^\dagger > \delta_c^* \) implies that the developer constraint is always satisfied by the platform sponsor’s choice.

To establish the claim with respect to \( \sigma \), apply the steps used in Proposition 1 to the system of equations including costs to produce the following pair of implicit functions.
\[ \sigma_c^+ : \alpha(vy_1 - \frac{1}{\alpha}cy_1^{1/\alpha}) + \delta_c^+\alpha^2(vy_2 - \frac{1}{\alpha}cy_2^{1/\alpha}) = 0 \]  \hspace{1cm} (13)

\[ \sigma_c^- : \alpha(py_1 - \frac{1}{\alpha}cy_1^{1/\alpha}) + \delta_c^\alpha^2(py_2 - \frac{1}{\alpha}cy_2^{1/\alpha}) = 2\sigma V \]  \hspace{1cm} (14)

Transform the first by mapping \( \delta_c^+ \) to \( \delta_c^- \) and the second by mapping \( p \) to \( v \). As second period surplus is always non-negative, the welfare and profit constraints are easily sorted.

\[ \sigma_c^+ : \alpha(vy_1 - \frac{1}{\alpha}cy_1^{1/\alpha}) + \delta_c^+\alpha^2(vy_2 - \frac{1}{\alpha}cy_2^{1/\alpha}) = -\kappa_1 < 0 \]  \hspace{1cm} (15)

\[ \sigma_c^- : \alpha(vy_1 - \frac{1}{\alpha}cy_1^{1/\alpha}) + \delta_c^-\alpha^2(vy_2 - \frac{1}{\alpha}cy_2^{1/\alpha}) = \kappa_2 > 0 \]  \hspace{1cm} (16)

Where \( \kappa_1 = \alpha\Delta(\alpha vy_2 - cy_1^{1/\alpha}) > 0 \) and \( \kappa_2 = 2\sigma V + \alpha\delta vy_1 + \alpha^2\delta^2 vy_2 > 0 \). Under model assumptions, the first constraint binds always to the left of the second. In this case, producing \( \sigma_c^+ > \sigma_c^- \).

We observe that the greater the share of downstream innovation captured by the platform sponsor, the greater is the incentive to open. The Nash share is \( s = \frac{1}{2} \) but, more generally, for \( s \in [0, 1] \), the platform sponsor’s constraint moves with \( \frac{1}{s}\sigma V \), which falls weakly toward the constraint of the social planner as \( s \) rises. This parallels results elsewhere in the literature: internalizing downstream innovation causes the owner of an upstream innovation to behave more like a social planner.

Interestingly, this also shows that higher costs cause the social planner to behave more like the proprietary sponsor. As total costs rise, \( \Delta \) falls (as profit terms tend toward 0) such that the choice of the social planner increasingly resembles that of the platform sponsor.
2.2.1 Technological Uncertainty

Since innovation can involve risk, we ask whether technological uncertainty influences the choice of openness and time to bundle. Let the probability of technical success be given by $\omega$ (thus “technological uncertainty” is $\rho = 1 - \omega$). Further, to balance risk and reward, allow output from riskier innovations to rise conditional on their success. Then, first period production is given by the random variable

$$Y_1 = \begin{cases} 
\frac{k}{\omega}(\sigma V)^{\alpha} & \text{with probability } \omega, \\
0 & \text{with probability } 1 - \omega.
\end{cases}$$  \hspace{1cm} (17)$$

This formulation assumes that in industries where technical success is difficult, i.e. $\omega$ is low, such success is highly rewarded.

Expected first round innovation is given by $E(Y_1) = k(\sigma V)^{\alpha}$ and variance is given by $Var(Y_1) = \left(\frac{1-\omega}{\omega}\right) k^2(\sigma V)^{2\alpha}$. Although the expected value of production is independent of technical risk, the variance of production increases with decreasing probability of technical success. In the limit, as $\omega \to 1$, we retrieve the original model with zero variance.

Similarly, provided that first period innovation was technically successful, second period production is given by the random variable

$$Y_2 | \text{success in period 1} = \begin{cases} 
\frac{k}{\omega}(y_1)^{\alpha} & \text{with probability } \omega, \\
0 & \text{with probability } 1 - \omega.
\end{cases}$$

The unconditional, time zero, production in the second period is given by:

$$Y_2 = \begin{cases} 
(\frac{k}{\omega})^{\alpha+1}(\sigma V)^{\alpha^2} & \text{with probability } \omega^2 \\
0 & \text{with probability } 1 - \omega^2.
\end{cases}$$  \hspace{1cm} (18)$$

14
The unconditional expected value of second stage production at time zero is 
\[ E(Y_2) = \omega^{1-\alpha}k^{1+\alpha}(\sigma V)^{\alpha^2} \] 
with variance 
\[ Var(Y_2) = \left(\frac{1}{\omega^{2\alpha}}\right)k^{2+2\alpha}(\sigma V)^{2\alpha^2}. \] 
Again, as \( \omega \rightarrow 1 \), we retrieve the original model with zero variance. Since \( 0 \leq \alpha \leq 1 \), the value of the second stage production is increasing in the likelihood of technical success \( \omega \) and therefore decreasing in variance. Low likelihood of technical success (i.e. low \( \omega \), high \( \rho \)) does not negatively affect the value of first stage innovation because innovation is more valuable if it is difficult to achieve, but it does negatively affect the value of second stage innovation because, for a second stage to exist, the first stage must be successful.

With these definitions, the platform sponsor profit function becomes:

\[ E(\pi_p) = V(1-\sigma) + \frac{1}{2}v(1-\delta)k(\sigma V)^{\alpha} + \delta \frac{1}{2}(1-\delta)k^{1+\alpha}(\sigma V)^{\alpha^2}\omega^{1-\alpha} \]  

(19)

Propositions 1 and 2 continue to hold but with \( y_1 \) and \( y_2 \) replaced by \( E(Y_1) \) and \( E(Y_2) \).

We summarize these implications in the following result.

**Proposition 4** Holding all else constant, greater technological uncertainty reduces platform openness and innovation, and increases the amount of time sponsors delay bundling and collect royalties. Increasing \( \rho \) implies that \( \sigma^* \) and \( Y_2 \) fall, while \( t^* \) rises.

Comparative statics are easy to evaluate. The effect of increasing technical success \( \omega \) goes in the same direction as increasing output \( Y_2 \). Increasing \( Y_2 \) increases both \( \sigma^* \) and \( \delta^* \). Therefore we can conclude that greater technical uncertainty (i.e. increased \( \rho \)) decreases the optimal choice of how much to open the platform. Also, because subsequent innovation entails more risk, the sponsor prefers to collect royalties \( t^* \) longer rather than gamble on innovation from bundling sooner.
2.2.2 Developer Number and Competition

To this point, the model has effectively assumed a single developer. How does increasing the number (or size) of developers and introducing developer competition affect platform sponsor choices for \( \sigma^* \) and \( t^* \)? Increasing the number of developers \( N > 1 \) raises output in each period such that \( \tilde{y}_1 = Ny_1 \) and \( \tilde{y}_2 = N^{1+\alpha}y_2 \). Increasing the intensity of developer competition softens prices such that \( \tilde{p} = \gamma v(1 - \delta) \) with \( 0 \leq \gamma < 1 \). More developers and more intense competition then have the following effects.

**Corollary 2** Increasing the size of the developer pool increases \( \sigma^* \) but decreases \( t^* \). Increasing competitive intensity decreases both \( \sigma^* \) and \( t^* \).

**Proof.** The comparative statics results from Corollary 1 provide a straightforward demonstration. Let \( \tilde{k} = Nk \) and \( \tilde{v} = \gamma v \) being careful to interpret rising competition as reducing \( \gamma \).

Intuitively, increasing the number of independent developers increases platform openness because downstream innovation increases at a higher rate. On margin, openness becomes more profitable. Having more developers also decreases the amount of time that applications should remain closed. By folding applications into the platform sooner, it becomes marginally more profitable for the platform sponsor to provide more resources to more developers who reuse these innovations as input to subsequent production.

We can combine this result with that of the previous section to see that as more developers help reduce technical risk, optimal openness rises further. Consider that if each developer represents an additional chance at technical success (with probability \( \omega = 1 - \rho \)), then the risk of technical failure declines as \( 1 - \rho^N \). Equations, 17 and 18 then become

\[
\tilde{Y}_1 = \begin{cases} 
\frac{Nk}{1-\rho} (\sigma V)^{\alpha} & \text{with probability } 1 - \rho^N \\
0 & \text{with probability } \rho^N,
\end{cases}
\]  

(20)
\[
\bar{Y}_2 = \begin{cases} 
\left( \frac{N_k}{1-\rho} \right)^{\alpha+1} (\sigma V)^{\alpha^2} & \text{with probability } (1 - \rho^N)^2 \\
0 & \text{with probability } (1 - \rho^N)^{\rho^N}.
\end{cases} \tag{21}
\]

These imply that unconditional expected values become \( E(\bar{Y}_1) = \frac{1 - \rho^N}{1 - \rho} \bar{y}_1 \) and \( E(\bar{Y}_2) = \frac{(1 - \rho^N)^2}{(1 - \rho)^{1+\alpha}} \bar{y}_2 \). The comparative statics are straightforward to evaluate. Both \( E(\bar{Y}_1) \) and \( E(\bar{Y}_2) \) rise in \( N \), thus increasing \( \sigma^* \). To evaluate the impact on time-to-bundle, replace \( \bar{y}_1 \) with \( \frac{\bar{E}(\bar{Y}_1)}{\bar{E}(\bar{Y}_2)} \) in Equation 6. The resulting expression is \( \delta = \frac{1}{2} \left( 1 - \frac{(1 - \rho)^\alpha}{1 - \rho^N} \frac{\bar{y}_1}{\bar{y}_2} \right) \). Since \( \frac{\bar{y}_1}{\bar{y}_2} = \frac{N}{N^{1+\alpha}} \frac{\bar{y}_1}{\bar{y}_2} \), we see that \( \frac{(1 - \rho)^\alpha}{1 - \rho^N} \) and \( \frac{\bar{y}_1}{\bar{y}_2} \) both decrease in \( N \), implying, respectively, that \( \delta^* \) increases and \( t^* \) decreases in \( N \).

This result is consistent with empirical research that finds handheld device platforms opened to more developers precisely to reduce the risk of technological innovation (Boudreau, 2008). For the same reason, social network platforms encourage developers to experiment with applications because “much remains unknown concerning preferences and technical approaches to social applications” (Hagiu & Boudreau, 2009; p. 11). Further, our model shows that, conditional on developer success, the platform sponsor profits by extending the royalty period for technically successful applications.

Competition among developers, however, has a different implication. Holding other factors constant, more intense developer competition reduces the Nash bargaining surplus available to the platform sponsor. This surplus goes instead to platform users, reducing the sponsor’s incentive to open the platform. Ironically, closing the platform has the effect of reducing developer output which reduces the value of applications royalties, as distinct from bundling them into the platform. Corollary 2 therefore shows that developer competition leads the sponsor to reduce the proprietary period.

That sponsors dislike developer competition stands in contrast to the standard result that platforms prefer to “commoditize complements” (Shapiro & Varian 1999; Gawer &
Cusumano 2002; Baldwin & Woodard 2008). The standard argument holds that the up-
stream platform prefers downstream competition to curb vertical pricing power and quantity
distortion. But this assumes complements exist. In a dynamic analysis, before downstream
innovation has occurred, the sponsor needs developers to *create* follow-on products. Thus
the sponsor prefers to give developers pricing power, lest they curb their downstream devel-
opment. This explains why platforms limit competitive intensity among developers of new
products via certification, royalty terms, and favorable directory placement (Boudreau &
Hagiu 2009). Alternatively, the platform sponsor might vertically integrate but must iden-
tify ex ante which developer innovations will succeed ex post. If the sponsor could identify
successful developers, then it might subcontract, a situation we analyze in Section 3.1. We
note simply that sponsor interest in downstream innovation also provides reason to prefer
(initially) less downstream competition.

### 2.2.3 Platform Competition

We now examine the effect of competition between platforms on the platform sponsor’s
optimal choice of $\sigma^*$ and $t^*$. In the same way that competition reduces developer pricing
power, platform competition reduces direct platform price from $(1 - \sigma)V$ to $(1 - \sigma)\lambda V$ with
$0 \leq \lambda < 1$. By varying $\lambda$, we see that increasing the intensity of platform competition has
the opposite effect of increasing the intensity of developer competition.

**Corollary 3** Increasing the intensity of platform competition increases both $\sigma^*$ and $t^*$.

**Proof.** To establish the first claim, substitute model primitives for output terms into
equation 4 from Proposition 1 and hold all else constant to show that the following equality
holds.

$$
\frac{b_1}{\sigma^{1-\alpha} \lambda} + \frac{b_2}{\sigma^{1-\alpha^2} \lambda} = 1
$$

(22)
Increasing competitive intensity by decreasing lambda implies increasing $\sigma$ in order to maintain the equality. To establish the second claim substitute constants for model parameters other than $\sigma$ into equation 6 from Proposition 2. The optimal choice of $\delta^*$ is governed by the following ratio.

$$ \delta^* = \frac{1}{2} \left( 1 - b \sigma^\alpha \frac{\sigma}{\sigma^\alpha} \right) $$

(23)

Given $0 < \alpha < 1$, we conclude that a larger $\sigma^*$ corresponds to a lower $\delta^*$ which implies a higher $t^*$.

Holding all else constant, greater platform competition reduces the direct platform surplus available to the platform sponsor. The sponsor’s incentive is therefore to open the platform in order to increase indirect profits from downstream innovation. Because the platform sponsor must take more of its profits from developer revenues, the platform sponsor also has a greater interest in maintaining developer price, which leads the sponsor to increase the proprietary period. The effect of platform competition is therefore to increase both openness and subsequent developer output. In terms of competition policy, the regulatory implication is that to achieve higher innovation, promote developer entry but not developer competition. Instead, promote platform competition which motivates sponsors to open and seek growth. This directly parallels empirical findings. Based on case studies of IBM, Sun Microsystems, and Apple, West (2003) concluded that sponsors generally prefer the higher rents from proprietary governance unless their platforms face significant pressure from rival platforms. We examine how this interacts with private subcontracting and property rights next.
3 Alternate Organizational Forms

In this section, we examine alternate ways to organize for innovation including the decision to subcontract instead of license openly and the decision of developers to cooperate rather than bargain with the platform sponsor.

3.1 Open Innovation vs. Private Subcontracting

Up to this point, we have assumed that firms rationally open their platforms to seek innovation. The cost of openness, however, is that the sponsor sacrifices profits on assets he could otherwise sell. Firms can avoid this cost by subcontracting directly with developers. Keeping the platform closed converts direct platform profits to \( V(1 - \sigma) \mid_{\sigma=0} = V \) Further, direct negotiation has the added benefit that the sponsor can share access to the full technology embedded within the platform, sharing only with subcontractors and no one else. This parallels Apple’s strategy of sharing with a small developer pool and producing a tightly integrated system (Boudreau 2007). Sharing only privately with subcontractors and giving them full access increases their output to \( y_1 = \frac{1}{2}pk(\sigma V)^{\alpha} \mid_{\sigma=1} \) and \( y_2 = \frac{1}{2}\delta pk^{1+\alpha}(\sigma V)^{\alpha^2} \mid_{\sigma=1} \). With these two benefits, modified platform profits become

\[
\pi_{sub} = V + \frac{1}{2}pkV^{\alpha} + \frac{1}{2}\delta pk^{1+\alpha}V^{\alpha^2}.
\]

In contrast, the virtue of open licensing is broader participation and increased user value. Mechanisms by which openness might increase platform participation or willingness to pay include ability of users to modify open systems, transparency and lack of “spyware,” free redistribution, openness as a commitment to low price analogous to second sourcing (Farrell
& Gallini, 1988), horizontal organization (Farrell et al., 1998), and lack of negotiation costs. This last attribute is especially salient for developers of novel applications who risk disclosing their ideas by identifying themselves or their applications to the platform sponsor (Bessen & Maskin, forthcoming). Recent work on “two-sided” networks (Rochet & Tirole, 2003; Parker & Van Alstyne, 2005) also demonstrates how subsidizing a developer community, as with $\sigma V$, increases platform value to and participation of a user community. Reciprocally, broadening the user base attracts a larger developer base. If platform development costs are relatively fixed and independent of the number of firms involved in downstream development, then the platform sponsor should be able to reduce R&D costs for creating the next generation platform by opening to complementary developers. Also, competition among developers may result in survival-of-the-fittest complements. Finally, open processes for jointly developing technologies invite ongoing feedback, which can yield higher quality products (Chesbrough, 2003; West, 2006). For a variety of reasons, openness can increase both value and participation.

The question we address in this section is when the benefits of openness outweigh the benefits of subcontracting. With minor modifications, we can analyze when openness modifies intrinsic application value $\bar{v} = Mv$ and when it modifies developer participation such that output moves with $\bar{y} = My$. Respectively, these two changes modify sponsor payoffs as follows:

$$\pi_{\text{open value}} = V(1 - \sigma) + \frac{1}{2}Mp y_1 + \frac{1}{2} \delta M p y_2(y_1)$$

$$\pi_{\text{open output}} = V(1 - \sigma) + \frac{1}{2}pM y_1 + \frac{1}{2} \delta pM y_2(My_1)$$

The result is that increased willingness-to-pay and increased participation can each justify
an open platform relative to closed subcontracting. The latter has a larger effect due to the compounding effects of production. More formally, we provide a strong bound.

**Proposition 5** If the platform sponsor values second period production, an open platform is more profitable than a closed subcontract whenever the multiplier on value is

\[ M > \frac{1}{\sigma^{2} + \frac{\sigma V}{\delta y_{2}}} \]

Measured in terms of participation, the condition is

\[ M^{1+\alpha} > \left( \frac{1}{\sigma^{2} + \frac{\sigma V}{\delta y_{2}}} \right) \]

**Proof.** The platform sponsor prefers openness when \( \pi_{\text{open value}} > \pi_{\text{sub}} \) or, after subtracting and grouping terms, when \( \sigma V < \frac{1}{2}\delta pk V^{\alpha}(1 - M\sigma^{\alpha}) + \frac{1}{2}\delta pk^{1+\alpha} V^{2}(1 - M\sigma^{2}) \). When the sponsor values second period production, the second righthand term exceeds the first. Thus a stronger bound on the inequality is \( \sigma V < \frac{1}{2}\delta pk^{1+\alpha} V^{2}(1 - M\sigma^{2}) \). Algebraic simplification yields \( \frac{\sigma V}{\delta pk^{1+\alpha} V^{2}} < M\sigma^{2} \). Division, then substituting for the definition of \( y_{2} \) provides the necessary expression. An identical sequence of steps for \( \pi_{\text{open output}} > \pi_{\text{sub}} \) that accounts for the compound effect of technology produces the second expression.

Opening the platform becomes more attractive (i) as the subsidy \( \sigma V \) falls (ii) second round output \( y_{2} \) grows, and (iii) technology \( \alpha \) improves. This proposition argues for decentralized innovation when user-developer network effects rise far enough. Note that the decentralized innovation is achieved without bargaining costs. A default open contract with \( \sigma > 0 \) gives developers an option to enter the market for any fixed costs up to the amount they can recover, and without current period disclosure to the platform author. They need not risk disclosing their idea to the monopsonistic platform author who could potentially appropriate its value.

**3.2 Cooperation in the Absence of Platform Control**

The ability to reuse material from one application in the development of another raises the prospect that developers can reciprocally contribute to one another’s forward development.
After all, access to a richer pool of application resources fosters richer application development. In effect, the platform sponsor appropriates developer resources at time \( t^* \) in order to make them available to other developers via the platform. Is “confiscation” necessary?

To analyze this problem, we consider the outcomes from cooperation versus defection with the former interpreted as contributing to the common resource pool and the latter means withholding resources in order to charge for them. The four strategies we consider are (i) cooperate, cooperate (\( CC \)) where the first position denotes the strategy of an individual developer and the second position denotes the action of the remaining developers, (ii) defect, cooperate (\( DC \)), (iii) cooperate, defect (\( CD \)), and (iv) defect, defect (\( DD \)). Denote \( \pi_{d_i}^{CC} \) as the profit that an individual developer makes when it cooperates and all other developers cooperate. The profits from the remaining three strategies are denoted similarly.

Individual developer profits differ in two ways. First, individual developers explicitly consider the number \( N \) of other applications apart from their own. Second, uncooperative developers can recover the revenues in the tail of the distribution \( t > t^* \). These changes yield the four strategies with surpluses as given in Table 1 and the proposition below.

**Table 1:** Surplus from the four strategies available to developers.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>( T_1 ) Own</th>
<th>( T_2 ) Other</th>
<th>( T_2 ) Own</th>
<th>( T_1 ) Tail</th>
<th>( T_2 ) Tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi_{d_i}^{CC} )</td>
<td>( \frac{1}{2}v(1-\delta)y_1 ) + ( \frac{1}{2}v\delta^2N^\alpha y_2 )</td>
<td>( \frac{1}{2}\delta v(1-\delta)y_2 ) + ( \frac{1}{2}v\delta y_1 ) + ( \frac{1}{2}v\delta^2 y_2 )</td>
<td>( 0 ) + ( \frac{1}{2}\delta v(1-\delta)y_2 ) + ( \frac{1}{2}v\delta y_1 ) + ( \frac{1}{2}v\delta^2 y_2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \pi_{d_i}^{DC} )</td>
<td>( \frac{1}{2}v(1-\delta)y_1 ) + ( \frac{1}{2}v\delta^2N^\alpha y_2 ) + ( \frac{1}{2}\delta v(1-\delta)y_2 ) + ( \frac{1}{2}v\delta y_1 ) + ( \frac{1}{2}v\delta^2 y_2 )</td>
<td>( 0 ) + ( \frac{1}{2}\delta v(1-\delta)y_2 ) + ( \frac{1}{2}v\delta y_1 ) + ( \frac{1}{2}v\delta^2 y_2 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \pi_{d_i}^{CD} )</td>
<td>( \frac{1}{2}v(1-\delta)y_1 ) + ( 0 ) + ( \frac{1}{2}v\delta(1-\delta)y_2 ) + ( \frac{1}{2}v\delta y_1 ) + ( \frac{1}{2}v\delta^2 y_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \pi_{d_i}^{DD} )</td>
<td>( \frac{1}{2}v(1-\delta)y_1 ) + ( 0 ) + ( \frac{1}{2}v\delta(1-\delta)y_2 ) + ( \frac{1}{2}v\delta y_1 ) + ( \frac{1}{2}v\delta^2 y_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Proposition 6** Among developers, [Defect, Defect] constitutes a dominant pure strategy Nash equilibrium.

**Proof.** We show a prisoners’ dilemma as follows. Direct comparison of \( CC \) and \( DC \) profits reveals that a profit-motivated developer prefers to defect when the other developers cooperate. That is \( \pi_{d_i}^{DC} = \pi_{d_i}^{CC} + \frac{1}{2}\delta v y_1 + \frac{1}{2}\delta^2 v y_2 \). The comparison of \( \pi_{d_i}^{DD} \) to \( \pi_{d_i}^{CD} \) is similar,
showing that profit-motivated developers defect. ■

Having established that profit-motivated developers will not, in the absence of enforce-
ment, cooperate by freely releasing their enhancements, we ask when a developer would
prefer to submit to a contract that would enforce the cooperative CC outcome. That is, we
compare the profits under DD to CC. First, note that in the case of DD, there is no open
stock release, so the user base and first and second period resource pools remain constant.
The only difference is that the developer has access to his own private stock as well as Ω₁.

**Proposition 7** If the platform sponsor chooses \( t^* < \infty \) there exists a contract committing
developers to give up their applications that makes them better off whenever \( N > 2^{\frac{1}{2}} \).

**Proof.** Comparing differential gains from \( \pi_{d_{CC}}^C \) to those in \( \pi_{d_{CC}}^C \), developer profits are
higher when \( \frac{1}{2} v \delta^2 k^{1+\alpha} N^\alpha (\sigma V)^\alpha > \frac{1}{2} v \delta \kappa (\sigma V)^\alpha + \frac{1}{2} v \delta^2 k^{1+\alpha} (\sigma V)^\alpha \). Since \( t^* < \infty \) it follows
that \( y_2 > y_1 \) so \( \frac{1}{2} v \delta^2 k^{1+\alpha} N^\alpha (\sigma V)^\alpha > 2^{\frac{1}{2}} v \delta^2 k^{1+\alpha} (\sigma V)^\alpha \). Rearranging produces an expression
\( N^\alpha > \frac{2 \delta}{1-\delta} \) whose right hand side rises strictly in \( \delta \). As \( \delta \) reaches its maximum at \( \frac{1}{2} \), further
manipulation produces the required result. ■

This proposition establishes that the total number of developers only needs to exceed
a small constant in order for the cooperative solution to produce greater surplus than the
uncooperative solution. This has strong implications for the role of the platform sponsor.
Essentially, the sponsor enforces a set of \( O(N) \) bilateral contracts binding developers to give
up their applications after a reasonable profit period in order that all developers may reuse
each others’ valuable resources. This not only economizes on \( O(N^2) \) transaction costs, it
increases the total surplus available to each individual developer.

A consequence of Proposition 7 is that developers can prefer governance by a platform
sponsor to that of an uncoordinated open standard. This resolves a classic “collective action”
problem (Baldwin & Woodard, 2008). In the absence of orchestrated governance, individual
incentives to profit maximize lead to Pareto inferior welfare in terms of innovation and profits. As the comparative statics of Corollary 1 show, the optimal timing of property rights can also depend on industry specific factors such as $k$ and $v$. If this is true, then an industry platform sponsor can craft more specific timing than a regulator whose rules apply across platforms. Relative to open standards and regulation, efficiency gains from platform sponsorship might therefore occur in coordination and in specificity. This allows innovation to adjust to the different “clockspeeds” of different industries.

The platform sponsor’s interest in efficient innovation has interesting real world application as a resolution to the problem of the “anticommons,” identified as the hold-up that occurs when too many different parties each can block downstream innovation because each has a conflicting yet interlocking property right (Heller & Eisenberg, 1998). Under a platform model, the platform sponsor unblocks later innovation by making earlier innovation available to all developers on a non-discriminatory basis - this illustrates ”openness” by our earlier definition. The sponsor uses its property right in the platform to grant access to developers conditional on securing the ability to bundle enhancements into future versions of the platform. Proposition 7 shows that far from encouraging developers to avoid the platform, bundling their applications can make them better off over multiple cycles of innovation. The sponsor’s self-interest in platform innovation motivates it to shepherd the platform much as if it were a social planner. Sponsor interest in downstream royalties, in fact, encourages it to delay bundling longer than would a social planner as the sponsor does not directly participate in consumer surplus.

4 Extensions

It is worthwhile examining the robustness of our analysis to changes in assumptions. Major assumptions include, (1) a point estimate of consumer value, (2) a Cobb-Douglas production
model, and (3) a one period useful lifetime for open platform stock and developer applications. Dynamics are captured using a two-period model.

Clearly, and consistent with other papers in the literature, we assume point mass consumer demand for tractability. Consumers enjoy positive surplus in our model as a result of platform openness and finite property rights for developer output. Also, many information goods are sold in bundles, making a point mass estimate of average value a reasonable approximation (Bakos & Brynjolfsson, 1999). Bakos and Brynjolfsson (1999), clarified by Geng, Stinchcombe, and Whinston (2005), show that the standard deviation of the item values in a bundle can be made arbitrarily small by aggregating additional goods into the bundle. Adding multiple features to a platform is easily interpreted using such an average value $v$.

The common assumption of Cobb-Douglas production is, again, made for tractability and allows for simple results expressed in terms of constant elasticity of output with respect to changes in technology. Similar conclusions can be obtained with alternate formulations but results are particularly elegant with the current specification. This model also introduces a novel choice parameter, contractual openness, which plays a central role.

Relaxing the assumption of a one period lifetime for platform and developer stock would complicate analysis but also strengthen results. If open platform stock stimulates production for additional periods, the increase in developer output also increases willingness to share the platform in order to share developer surplus. On margin, openness and decentralized innovation then become more valuable.

5 Conclusions

The analysis presented here argues for several simple but important results. First, we show that platform sponsors can control downstream innovation, increasing profits through an
optimal choice of openness. They find it privately rational to stimulate production by others even at the cost of sacrificing direct platform sales. The openness condition is determined by the increase in developer output relative to subsidy cost as mediated by the elasticity of output across periods. This result clarifies strategies of firms that have relaxed platform control after building valuable brands.

Second, analogous to periods of patent protection, we identify conditions for a finite exclusionary period. In our model, this represents the time during which downstream developers can charge for new applications before the sponsor folds these enhancements into the open platform. Platform envelopment of first period innovations should occur at a time determined by the point at which second period developer output exceeds first period output. If second period output is smaller, then it is never optimal to fold developer enhancements into the platform as this reduces first period surplus.

Managers of proprietary platforms face a challenge as they try to manage the platform ecosystem: they must balance current and future profit. Applications developers can view them as acting too aggressively when they fold applications into the core platform. On the other hand, if managers are too slow, then consumers will face an increasingly complex task in trying to integrate disparate applications into end-user systems.

Third, we show that a social planner chooses to release a greater portion of the platform and forces earlier disclosure of developer production. However, increasing costs leads a platform sponsor to behave more like a social planner.

We also analyze the size of the developer pool and the intensity of competition among developers and platforms. More developers leads to a more open platform and also decreases the time until new features become part of the platform. In contrast, increased developer competition reduces openness because it reduces surplus available to the sponsor. More competition also shortens the proprietary period because new value comes relatively more from new production than from existing sales. Increasing platform competition has the
opposite effect. Platform sponsors have less direct profit and therefore prefer to increase developer revenues through a more open contract with a longer proprietary period.

Fourth, the model provides conditions for choosing between competing contract types. To promote sequential innovation, a sponsor can choose closed developer contracts that do not sacrifice platform profits or open contracts that stimulate greater developer participation. Open contracts that lead to decentralized innovation are increasingly preferred when the subsidy cost is smaller, developer output larger, or technology superior. These results are achieved without appeal to transaction costs, which should intrinsically favor open contracts that are simply default offers requiring no negotiation.

Fifth, we demonstrate a prisoners’ dilemma where developers individually refuse to open their applications even as they prefer every other developer open theirs. Given a sufficiently large developer pool, however, all developers are better off submitting to a contract forcing them to open their applications. The reason is that subsequent output can build from a larger pool of initial input, leading to higher total surplus. The platform sponsor must enforce such contracts not only for benefit of the platform but of the developers themselves, a role not unlike that of a social planner. This result is of particular importance for regulators and platform systems designers. In order to maximize the value creation potential of a platform ecosystem, the platform sponsor must have a longer tenure than the developers who build upon it.
References


6 Appendix

6.1 Derivation of Comparative Statics

We use the envelope theorem to establish the effect of exogenous parameters on the platform’s optimal choice of $\sigma$ and $\delta$. For example, by the envelope theorem, $\text{Sign} \left[ \frac{\partial g}{\partial k} \right]$ is equal to $\text{Sign} \left[ \frac{\partial^2 \pi_p}{\partial \sigma \partial k} \right]$. 

6.1.1 The effect of $V$ on $\sigma$ is negative

To determine the sign of $\frac{\partial \sigma}{\partial V}$, take the partial derivative of equation 5 with respect to $V$ to get

$$\frac{\partial^2 \pi_p}{\partial \sigma \partial V} = -\frac{k^{\alpha + 1}V^\alpha \alpha^2 (\delta - 1) \delta \sigma^\alpha + kV^\alpha \alpha^2 (\delta - 1) \sigma^\alpha + 2V \sigma}{2V \sigma}.$$

Substituting price, output, and elasticity terms for the model primitives, this can be rearranged as

$$\frac{(1/2)pV_1 \eta_1^2}{\sigma V} + \frac{(1/2)\delta pV_2 \eta_2^2}{\sigma V} - 1.$$  \hspace{1cm} (24)

Note that this differs from Equation 4 only by a square on the elasticity terms. Since both elasticities are less than one, this expression must be negative.

6.1.2 The effect of $v$ on $\sigma$ is positive

Calculate the cross partial derivative of profit as

$$\frac{\partial^2 \pi_p}{\partial \sigma \partial v} = \frac{k\alpha(1 - \delta)}{2\sigma} \left( k^\alpha \alpha^2 M^\alpha \delta \sigma^\alpha + V^\alpha \sigma^\alpha \right).$$

Note that all terms are positive.

6.1.3 The effect of $k$ on $\sigma$ is positive

Take the partial derivative of equation 5 with respect to $k$ to get

$$\frac{\partial^2 \pi_p}{\partial \sigma \partial k} = \frac{1}{2}p\sigma^{-1}V^\alpha + \frac{1}{2}\delta pk^\alpha \sigma^\alpha - 1V^\alpha.$$

Since all terms are positive, we conclude that $\frac{\partial \sigma}{\partial k}$ is also positive.
6.1.4 The effect of $\alpha$ on $\sigma$ is indeterminate

Take the cross partial, combine log terms and substitute $y_1$ and $y_2$ for the primitives to get the following equation.

$$\frac{\partial^2 \pi_p}{\partial \sigma \partial \alpha} = \frac{v(1-\delta)((\alpha \log(V\sigma) + 1)y_1 + \alpha \delta (\alpha \log (kV^{2\alpha}\sigma) + 2)y_2)}{2\sigma}$$

The sign of $\log V\sigma$ and $\log kV^{2\alpha}\sigma$ depends upon specific parameter values. Since $0 < \sigma < 1$, $0 < k < \infty$, and $0 < V < \infty$ the sign of $\frac{\partial^2 \pi_p}{\partial \sigma \partial \alpha}$ is indeterminate.

6.1.5 The effect of $V$ on $\delta$ is negative

We calculate the following cross partial derivative.

$$\frac{\partial^2 \pi_p}{\partial \delta \partial V} = \frac{v\alpha(k^{\alpha+1}MV^{\alpha^2}\alpha(1-2\delta)\sigma^{\alpha^2} - kV^{\alpha}\sigma^{\alpha})}{2V}$$

Substitute $y_1$ and $y_2$ for model primitives to get

$$\frac{\partial^2 \pi_p}{\partial \delta \partial V} = \frac{v\alpha(y_2\alpha(1-2\delta) - y_1)}{2V}.$$  

This is negative when $y_2\alpha(1-2\delta) < y_1$. Divide both sides by $y_2$ to express this as

$$\alpha(1 - 2\delta) < \frac{y_1}{y_2}.$$  

The optimal solution for $\delta$ requires that $\frac{y_1}{y_2} = (1-2\delta)$. Given $\alpha < 1$, we conclude that $\delta$ falls in $V$.

6.1.6 The effect of $v$ on $\delta$ is negative

The direct calculation of $\frac{\partial^2 \pi_p}{\partial \delta \partial v}$ returns the first order condition for $\delta$ and is otherwise inconclusive. So, we analyze the effect of $\sigma$ on $\delta$ to make a statement about the effect of $v$ on $\delta$. Take the optimal expression for $\delta$ (Equation 6) and substitute the model primitives for $y_1$ and $y_2$ to get the following expression.

$$\delta = \frac{1}{2} \left( 1 - k^{-\alpha}V^{\alpha - \alpha^2} \sigma^{\alpha - \alpha^2} \right)$$

Given $0 < \alpha < 1$ we conclude that $\delta$ falls in $\sigma$. Above, we establish that $\sigma$ grows in $v$. Therefore, $\delta$ falls in $v$.  

32
6.1.7 The effect of $k$ on $\delta$ is positive

We calculate the following cross partial derivative.

$$\frac{\partial^2 \pi_p}{\partial \delta \partial k} = \frac{1}{2} v \left( k^\alpha V^{\alpha^2} (1 + \alpha)(1 - 2\delta)\sigma^{\alpha^2} - V^\alpha \sigma^{\alpha} \right)$$

Multiply and divide by $k$ to restore expressions for $y_1$ and $y_2$.

$$v \frac{((1 + \alpha)(1 - 2\delta)y_2 - y_1)}{2k}$$

This is positive by the optimal solution to $\delta$, Equation 6.

6.1.8 The effect of $\alpha$ on $\delta$ is indeterminate

Take the cross partial, combine log terms and substitute $y_1$ and $y_2$ for the primitives to get the following equation.

$$\frac{\partial^2 \pi_p}{\partial \delta \partial \alpha} = \frac{1}{2} v \left( (1 - 2\delta) \log (kV^{2\alpha} \sigma) y_2 - \log (V \sigma) y_1 \right)$$

The sign of $\log V \sigma$ and $\log kV^{2\alpha} \sigma$ depends upon specific parameter values. Since $0 < \sigma < 1$, $0 < k < \infty$, and $0 < V < \infty$ the sign of $\frac{\partial^2 \pi_p}{\partial \delta \partial \alpha}$ is indeterminate.