Six Challenges in Platform Licensing and Open Innovation

Geoffrey PARKER & Marshall VAN ALSTYNE (*)
Tulane University and Boston University / MIT

Abstract: This article describes six common challenges of design, incentives, and governance that arise in establishing platform businesses. It also proposes solutions. It considers, for example, how to open a platform to decentralized innovation yet still earn a return; how to incorporate best-of-breed innovations from different sources while avoiding problems of multi-party hold-up; and how to encourage sources of good ideas to contribute those ideas despite the risk of losing them to owners of indispensible complements. We express these issues and solutions as a reduced set of tradeoffs useful for managing information and technology property.

Key words: licensing, open source, free software, dual licensing, platform, intellectual property.

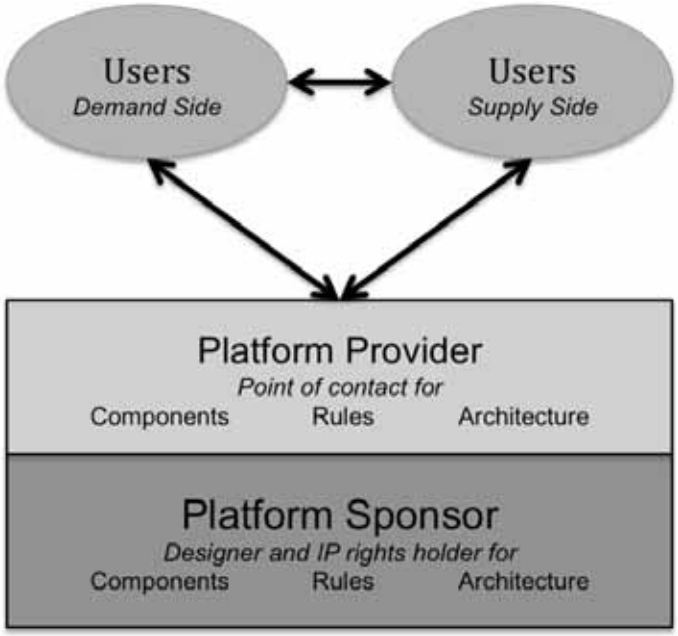
Information intensive, platform businesses account for an increasing share of the global economy. They also account for considerable innovation. The multi-party nature of platform ecosystems, however, implies that businesses must organize for decentralized innovation. In platform industries, even the biggest firms can no longer innovate alone. GAWER (2009) provides a useful definition:

> "Industry platforms are products, services or technologies that are developed by one or several firms, and which serve as foundations upon which other firms can build complementary products, services or technologies."

(*) Acknowledgements: We have benefited from the comments of many seminar and conference participants including those at the Workshop on Information Systems and Economics, the Institut d'Economie Industrielle, the Stanford Institute for Economic Policy Research, and the University of Michigan Law School. We have also benefited from conversations about intellectual property economics with many colleagues, including Jacques Cremer, Lawrence Lessig, Patrick Rey, Richard Stallman, and Jean Tirole. We are also thankful for the comments provided by two anonymous reviewers and are particularly thankful for the detailed criticism and feedback from Kevin Boudreau, Paul David, Annabelle Gawer, and Joel West. 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.

When describing platform businesses, we adopt the nomenclature introduced in EISENMANN, PARKER & VAN ALSTYNE (2009) and reproduce the figure below that shows the relationships in a platform-mediated network. Platform sponsors are responsible for the design and evolution of the platform system. Platform providers (who may also be sponsors) interact directly with platform users. Supply side users are the application developers who build on top of the core platform to extend its functionality. Demand side users consume platform and application resources, and interact with application providers via the platform (PARKER & VAN ALSTYNE, 2005; EISENMANN, PARKER & VAN ALSTYNE, 2006).

Figure 1 - Elements of a platform business network



Our purpose below is to describe six common challenges of design, incentives, and governance that arise in the establishment and management of platform businesses. These issues are developed from the perspective of a platform sponsor that holds platform intellectual property rights and can set the rules of engagement for participants in a platform ecosystem. In particular, we are interested in how the platform sponsor sets rules for application developers that wish to participate in the platform business. In addition, we compare the most commonly used software copyright licenses

and suggest that there might be alternate licenses that promote higher rates of innovation.

Increasingly, industries that were once characterized by slow rates of innovation are being restructured in ways that increase their "clockspeed" (FINE, 1998). For example, although there have been major improvements in electric power transmission and distribution technologies, the basic elements of the electric power system have remained unchanged since the days of Thomas Edison. That is about to change as utilities prepare to introduce a "Smart Grid" made up of new "applications" that promise improved real-time price response systems, more efficient delivery mechanisms, the integration of intermittent resources into distributed generation, increased options for energy-use management, and improved robustness to natural events such as hurricanes and ice storms. As a result, a once slowing-changing industry is about to undergo a rapid increase in rates of innovation.

To foster higher rates of innovation, the rules governing access and intellectual property must be carefully analyzed, designed, and enforced. This involves designing incentives for partners to participate and for partners to add value (BOUDREAU & HAGIU, 2009). Often, layers of innovation become part of the common platform available to users upon which others can further build; the process of placing innovations into the common platform is one of the central tasks facing a platform sponsor. Platforms are most likely to bundle application layers when similar features are provided by multiple developers and pricing power for any supplier is eroding. At this point, demand side users are faced with a coordination challenge that the platform can solve by folding these applications into the core while developers need to transition to more profitable applications [1]. Importantly, for platform sponsors to properly coordinate the developer community, they must have long lived propery rights that span multiple generations of developer applications. In Challenge 5 on free riding, we provide an explicit argument for how developers can be better off when a long-lived platform sponsor coordinates the ecosystem even when the sponsor forces the developers' applications into the common platform.

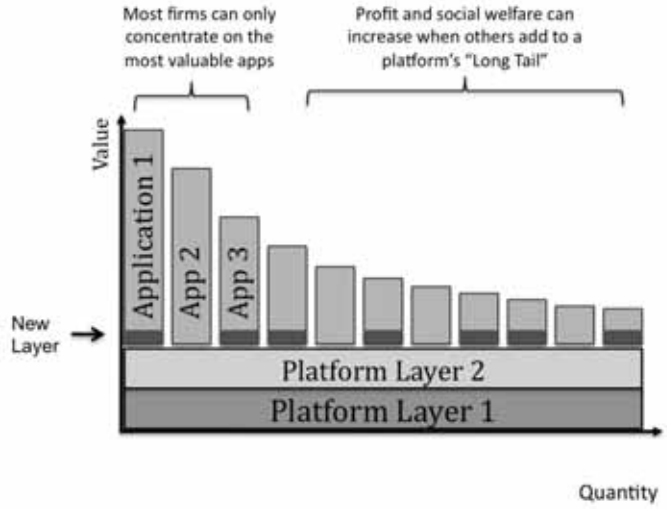
A platform's decision to bundle portions of an application into the common platform has an effect similar to forcing a developer to release code

[1] We thank Guido Jouret of CISCO for extensive conversations explaining the conditions under which firms should fold applications into their core platforms.

into the open domain. In both cases, pricing power is reduced or eliminated, although competition also erodes pricing power for any particular application. Platforms will wish to avoid deterring for-profit developers who generally wish to keep their particular layer closed. However, sponsors that delay moving application features into the core platform can stagnate and become less appealing to demand side users.

Figure 2 presents a graphical view of a platform system that allows for functionality to be extended through complementary applications. For a product or service to have platform potential, it must first perform a useful function and it must second allow unaffiliated parties to extend it in unanticipated ways (GAWER & CUSUMANO, 2002; BOUDREAU, 2008). The original iPod, offering nothing more than MP3 play, had limited functionality and extensibility and offered no developer access. In contrast, the iPhone, with color video, phone service, camera, wireless access, music play, and geolocation, was highly extensible and, eventually, offered developer access.

Figure 2 - Platform value can increase as 3rd party developers add applications. Over time, platform merges common features across applications into new platform layers



Other examples include process swapping (e.g. an operating system extended by applications), routing (e.g. telecomm-unications switching extended by call blocking) and hosting (e.g. software as a service extended by new customer programs). Many platform sponsors also provide applications (such as the Microsoft Windows platform and Word application)

and may choose to leave these applications out of the core platform, even over multiple generatations, as a way to price discriminate across applications and the platform. In our examples, we focus primarily on information and software, but we believe platform business models extend widely across traditional industries as network strategies become more widely adopted. Earlier work shows that leading firms have prospered by maintaining tight control over access to their platforms while others have prospered by opening access to third party contributions. Such distinct strategies suggest there are subtle answers to the question of how platforms should interact with supply side developers (WEST, 2003; EISENMANN, PARKER, & VAN ALSTYNE, 2006; 2009).

■ Intellectual property and innovation debate

Previous works on intellectual property have focused largely on standards and sequential innovation and have fallen largely into two camps - those favoring strong protection and those favoring weak. On one hand, economists and lawyers argue that incentives matter for innovation (LANDES & POSNER, 2003; WAGNER, 2003). Intellectual property law embodies this principle, granting temporary monopolies to authors and inventors as a stimulus to innovation. If too-strong property rights are truly bad for business, firms can simply choose not to exercise them (MERGES, 2008). On the other hand, recent developments have challenged the idea that proprietary systems call forth the most innovation. The challenge appears among other economists who assert that openness permits access and growth (SHAPIRO & VARIAN, 1999). Value also comes not from protection but from technological lead time (BESSEN & MASKIN, forthcoming). The challenge appears in another form among Free Software proponents, who argue for openness as a right, and among Open Source Software proponents, who argue for value created by peer review, reuse and unfettered redistribution (RAYMOND, 1999; DAVID, 2004). Once a property rights regime is established, firms must then grapple with the degree to which they should foster or limit competition among developers. Too much competition (such that oligopoly pricing power is eroded) implies that developers will stay away [2]. Even open source developers can avoid

[2] The number of firms necessary to erode profit margins is dependent upon many factors including the cost structures of each firm, the size of the total market, and the degree of differentiation between product and service offerings.

tackling problems for which they perceive too little credit (RAYMOND, 1999). But, too little competition implies disinterest, lack of critical mass, and a less vibrant platform. Below, we organize a handful of key papers in the literature by whether they argue for open or closed systems and the degree to which they argue that competition versus monopoly promotes innovation.

Table 1 - Intellectual property debate

Closed is better	*Free / Open is better*
Long but narrow patents (GILBERT & SHAPIRO, 1990)	Fundamental right of access (STALLMAN, 1992)
Infinitely renewable (LANDES & POSNER, 2003), long duration (WAGNER, 2003), or strong property rights (MERGES, 2008)	Collective production / Open science (BENKLER, 2002; DAVID, 2004)
Sequential innovation (GREEN & SCOTCHMER, 1995; CHANG, 1995)	Tragedy of the "AntiCommons" (HELLER & EISENBERG, 1998)

Table 2 - Innovation debate

Monopoly better promotes innovation	*Competition better promotes innovation*
"To promote progress in science and the useful arts", U.S. Constitution	No double marginalization (SPENGLER, 1950; MOTTA, 2004)
Competition reduces incentives to enter markets (SALOP, 1977; DIXIT & STIGLITZ, 1977)	Innovation occurs to "escape" competition (AGHION *et al.*, 2005). Imitation spurs innovation (BESSEN & MASKIN, forthcoming)

We note two key points about the literature in Tables 1 and 2 [3]. First, these papers are not all clear about the distinction between open access to a standard versus having the right to participate in defining that standard. In describing platform markets, GAWER & HENDERSON (2007) and GAWER & CUSUMANO (2008), emphasize that control rights are multi-layered. Openness is not all-or-nothing but has many shades of grey (WEST, 2006). In addition, most of the papers in Tables 1 and 2 do not distinguish between standards and platforms. The latter exhibit architectural control points where decisions regarding the grant of control rights to others must be made. These decisions can precipitate or preclude third party development. Platform sponsors must plan these multi-layered control rights, when to open parts of the platform, and when to fold parts of applications into new platform layers.

[3] We thank Annabelle Gawer for pointing out the need to clarify how the intellectual property and innovation literatures differ from the platform literature.

■ Common license access properties

Before describing our six challenges in platform design, incentives, and governance, we first list several of the licenses that are observed in the software domain to make some general comments about their properties. Below, we reproduce the license categorization from an internal Microsoft memorandum, the Halloween Document (VALLOPPILLIL, 1998). This table arranges licenses from the most restrictive end user license agreement (EULA) to licenses that force immediate disclosure of all derivative work (Linux/GNU).

The arrangement of licenses in terms of increasing access suggests that there might be a continuous representation of two key features. The first of these features is the degree to which the license opens the information asset. The second feature is the time after which a derivative innovation must be put into the open domain.

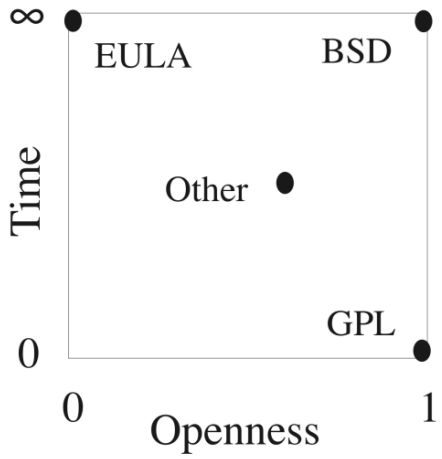
Table 3 - Halloween document (VALLOPPILLIL, 1998)

Type	*Free*	*Redist*	*Unlimited Use*	*Src. Avail.*	*Src. Mod.*	*Public Checkin*	*Viral*
Commercial (EULA)							
Trial Software	x						
Shareware	x	x					
Royalty Free Binary	x	x	x				
Royalty Free Library	x	x	x	x			
Open Source (BSD)	x	x	x	x	x		
Open Source (Apache)	x	x	x	x	x	x	
GPL-style Open Source (Linux/GNU)	x	x	x	x	x	x	x

Based on the license table above, we construct Figure 3 below and map three common licenses along our two key dimensions: the degree of immediate openness for the first round of innovation, and the length of time before derivative works must be released into the public domain. The licenses we map include a standard shrinkwrap software end user license

agreement (EULA), the Berkeley Standard Developer's (BSD) license, and the GNU Public License (GPL). Fully open source software, such as that released under the GPL releases everything, and requires subsequent code to be released immediately. In contrast, the Berkeley Standard Developer's (BSD) license releases everything but places no restrictions on subsequent disclosure. The Microsoft End User License Agreement (EULA), for example, does not disclose the underlying features of the software beyond user interfaces (disclosure after expiration of copyright, 95 years after publication, is effectively forever in economic terms). Interestingly, in this framework, many standard licenses appear as corner solutions. This suggests that there might be "Other" interior licenses that are socially superior. Such a license might be termed a flexible copyright, or meta license, that incorporates features of both proprietary copyright and open source copyleft. That is, such a license might require an intermediate degree of openness immediately and complete openness after an intermediate time.

Figure 3 - Licenses mapped by platform openness and time to release derivative works



PARKER & VAN ALSTYNE (2008) show conditions under which interior "Other" licenses dominate the three corner licenses along dimensions of platform sponsor profitability and social welfare. The key economic mechanisms are that opening more of the platform to subsequent rounds of innovation reduces immediate profit, but can create more innovation (and profit and social surplus) in future periods. However, as we argue below, if platform sponsors and developers are forced to release their innovations immediately, there is no profit motive. Conversely, if derivative works are

never released to the open domain (or are never folded into the core platform), then developers lose incentives to innovate in subsequent periods.

The key argument is that there exist economic reasons why a profit-maximizing firm would move from a strict proprietary license in the direction of an open source license. Conversely, there exist reasons why a welfare-maximizing social planner would move from a strict open source license in the direction of a proprietary license.

To maximize the long term amount of for-profit and free/open software, a platform sponsor should make available a portion of the code in order that third party developers can enhance and extend it. In the vernacular of the open source movement, many users need to "scratch an itch" and cannot adapt code without access to the original sources (RAYMOND, 1999; LERNER & TIROLE, 2002; VON HIPPEL & VON KROGH, 2003). Using property rights in the platform, the sponsor can then cause subsequent innovation to also become free/open by bundling the best features into the platform. The time horizon for optimal bundling balances the interests of ecosystem partners – long enough that partners make money but soon enough that newly opened features lead the platform to further evolve. This process of opening and bundling creates a stream of derivative work in which the original platform sponsor may exercise an interest. Managing that interest, whether for public welfare or personal profit, is the subject of the tradeoffs below.

■ Platform design issues

We briefly set forth below six design challenges that arise in developing licenses that mix proprietary and open source elements. The goal is to explore the possibility that a new license might capture the best of proprietary incentive systems and open source free distribution with peer review. This idea parallels the patent system wherein a right to exclude is briefly conferred.

Challenge 1: Open source can curb developer incentives

Issue: While standard open source licenses do not require developers to license their enhancements at cost, that is their economic effect. If

developers must give users both the enhanced code and the right to redistribute, then developers must compete with perfect zero-marginal cost copies of their own goods. Such markets cannot sustain positive prices above transactions costs on the good itself. Although developers have attempted business models based on indirect compensation, such as services, they have not sustained prices that reflect the economic value of their innovations (LERNER & TIROLE, 2002). For economies not based on gift exchange (LAKHANI & VON HIPPEL, 2003), this leads to reduced social welfare characterized by under-investment in innovation.

Proposed solution: One mechanism is to give developers pricing power in their enhancements by delaying the time until the right to redistribute a copy vests with users. This leads to declining pricing power over time as the free/open period approaches. The length of delay in rights to redistribute, i.e. a proprietary period of an enhancement, should set the area of the demand curve under developer price proportional to the size of investment one wants to call forth while also accounting for the opportunity cost of the developer's time. The key idea is that by allowing 3rd party developers to maintain certain rights in their own enhancements, the platform sponsor can attract deeper and more sustained complementary investments.

Note that the offer of a proprietary period places no obligation on the developer to use it. Rather it presents an option to be used at the developer's discretion and he or she may choose the traditional open source gifting role, contributing back to the common code base at time zero or at any time that simply recovers costs. It does, however, create an economic incentive for those who might wish to profit from their own innovation.

Challenge 2: Information asymmetry in proprietary licenses

Issue: For many proprietary licensing opportunities, the developer of a new idea may need to disclose his or her concept to the platform author in order to obtain both access to the source code and permission to create a derivative work. As owner, however, the platorm author is under no obligation to license and, having learned of the opportunity, may simply refuse to license and then exploit the opportunity without the developer. This is the monopsony problem in which the existence of only one possible buyer creates severe problems of hold-up and inefficiently modest levels of investment, innovation, and trade (ARROW, 1962).

Proposed solution: A virtue of free/open source software licensing is the offer of a default contract requiring no review by the platform sponsor [4]. Any person with an idea can anonymously secure both access to source code and permission to enhance it without disclosing private knowledge. The solution is thus to offer a public default contract, available to anyone, provided that developers meet other reasonable terms of the license.

Note that another version of this solution exists already in the form of applications program interfaces (APIs). Much proprietary code exposes APIs to programmers in order that developers can execute function calls on other software. The current proposal is to extend this proprietary model in the direction of open source such that subroutines can be modified and reused and not merely accessed or called. In addition, this also helps to resolve the developer suspicions of "hidden" APIs whereby platform sponsors might favor internal developer teams (ELZINGA, EVANS & NICHOLS, 2001). GAWER & HENDERSON (2007) document the extensive steps taken by Intel's Architecture Lab to convince developers that they would have the same access to new personal computer platforms as Intel's internal technology developers.

Challenge 3: Code forking and incompatibility

Issue: Certain licenses, notably BSD, permit code versions that are incompatible. This practice, known as forking, happens when developers create incompatible versions from a common code base. It arises particularly in cases where developers are allowed to keep their code separate into perpetuity and thus have an economic incentive. Forked code can also be used to fragment a platform as Microsoft once attempted with Java. In open source projects, teams can disagree over goals and objectives and may pursue independent paths. In a simple two-party example, party A may be unwilling to share control while party B may be unwilling to use party A's changes. Either action can cause a fork in the code base which can reduce consumer surplus via a reduction in standardization and network effects [5].

Proposed solution: Require developers to license their enhancements back to the platform sponsor in order that improvements can be folded back into the common code base at the appropriate time. Note that developers

---

[4] In the law literature, this is often called a "standard form contract".

[5] Joel West provided valuable clarification on how code base forks arise.

can continue to sell independently if they wish but it becomes economically unattractive. The sponsor should then use economic principles of bundling to discourage incompatible versions. Bundling features creates substantial barriers to entry (NALEBUFF, 2004). The key insight is that once a feature becomes a standard platform component, a product competing for the same users must offer differentiated value in order to merit paying for what would otherwise be redundant functionality.

Although bundling creates barriers to competing product entry, these are not insurmountable. Of interest is the fact that it remains contractually feasible to fork the code base. In order for this forking strategy to succeed, however, the economic value of the new offering must be sufficiently great – a developer must add sufficiently valuable functionality, often as part of a competing bundle – that the forked alternative can survive in the market. This permits a radical innovation to evolve while forcing it to pass a threshold test of value that has not simply been contractually excluded.

Note that the problem of versioniong also arises in the case of "dual licensing," the practice of offering simultaneous open and closed versions of the platform. The closed version is what the sponsor can charge for but the open version has the higher rate of potential growth from third party features. If the sponsor were to limit third parties from redistribution, then it forfeits the full benefits of open innovation. But if it does not limit redistribution, the open version can develop valuable new features not available in the closed version. The sponsor can seek to capture and sell the value of third party development [6] but this can also reduce developer contributions as they resist working for free. In contrast, if developers can charge for the value of their contributions, this simply restores developer pricing power as suggested in Challenge 1.

Challenge 4: Quality, Competition and value chain hold-up

Issue: Technology markets frequently exhibit either "component competition" or "systems competition" (FARRELL, MONROE & SALONER, 1998). In the former, an innovator can compete on the basis of specializing in a uniquely low cost or high value part where they have an advantage. In the latter, an integrator competes by offering a high value collection without necessarily offering the best-of-breed or lowest cost for any part. The

6 In the law literature, such contract clauses are often called "grant backs".

economic consequence is that specialized providers suffer hold-up by other bottleneck suppliers in the value chain, while integrators suffer technological obsolescence and strategic defection by specialist suppliers from whom they purchase. The crux is that individual firms cannot perpetually provide the best of every part but fragmentary rights distributed among multiple innovators create welfare losses through multiparty bargaining (HELLER & EISENBERG, 1998).

Proposed solution: Software permits near zero marginal cost transfer of the enabling technology throughout a value chain via access to source code. A successful mechanism might therefore offer default rights to the community of developers after the developer of an enhancement has been compensated both for the costs of innovating and the opportunity cost of effort. This simply occurs through the termination of the proprietary period and the commencement of the open/free source period. The most successful or "best-of-breed" enhancements become part of the common code base while reducing economic distortions caused by multiparty bargaining. Termination of the proprietary period then gives every developer a "right to merge" the best of breed components.

Of interest to platform sponsors is the problem of how to identify best of breed components. No one should assume that every contribution is worthy of platform association. The Atari Gaming system, for example, suffered significant reputation harm when, after opening to outside development, poor quality games flooded its market. One solution is to track the most popular items. Another is for the platform sponsor to invest in certifying component quality. NTT DoCoMo uses the limited window space on its mobile phones to feature a dynamic list of the most popular items. Cisco encourages broad development on top of its networking products but certifies only a limited number in order to maintain exclusivity and prevent brand deterioration.

Challenge 5: Free riding by 3rd party developers

Issue: The creation of a public good, one that is nonrival and nonexcludable, simultaneously introduces incentive compatibility problems due to free riding. In this case, a third party developer might wish to invest in an enhancement but also to dishonor the principle of releasing source code upon expiration of the proprietary period.

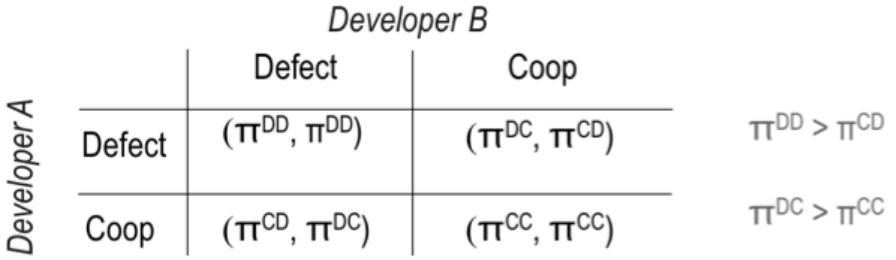
Proposed solution: The sponsor's task is to offer developers sufficient value in affiliating with its platform that developers choose to create a

derivative work in preference to incurring the cost of developing an independent platform. This implies that the offer of developer profits collected during the proprietary period must be greater than the developer could earn independently, accounting for (i) the cost of independent platform development and (ii) the lower cost of reusing common code.

Note that another common problem of public goods, over-grazing or the "tragedy of the commons" does not arise in the case of software due to zero marginal cost reproduction. PARKER & VAN ALSTYNE (2008) show in a modeling framework that there is a prisoner's dilemma in the disclosure game among developers (see Figure 4). The Nash equilibrium is (defect, defect) where defect means to keep one's innovation closed while benefitting from others' disclosures. Cooperate means to allow one's own innovation to become open after a time. Although the Nash equilibrium is for developers to defect, this is not the Pareto optimum (PARKER & VAN ALSTYNE, 2008).

To make money, developers can individually refuse to open their applications even as they prefer every other developer to open theirs. Given a sufficiently large developer pool, however, developers are collectively better off submitting to a contract forcing them to open their applications at a future date. The reason is that subsequent output can build from a larger pool of initial input, leading to higher total surplus.

Figure 4 - Prisoner's dilemma in developer disclosure game



The platform sponsor must enforce such contracts not only for benefit of the platform but of the developers themselves, a role similar to that of a social planner. In effect, the platform sponsor's own self interest causes the sponsor to seek a solution to the public goods problem – namely underprovision of new information due to non-cooperation. This result is of particular importance for regulators and platform systems designers. In order to maximize the value creation potential of a platform ecosystem, platform

sponsors must have longer tenure than the developers who build upon their platforms.

Finally, we note the existence of norm-based exchanges that provide non-contractual solutions to the free-rider problem. For example, FAUCHART & VON HIPPEL (2008) describe how strong cultural norms protect the intellectual property (recipes) of French chefs. Violations of the norms are punished with sanctions that include loss of status in the community and loss of future access to the community's resources.

Challenge 6: Platform sponsor rent seeking

Issue: If the platform sponsor maintains a key complement as inaccessible proprietary code, then that sponsor can potentially appropriate the value of future enhancements via price hikes. Once the developers' enhancements become part of the core platform, they are relatively free, but demand side users must still purchase the indispensible complement to receive the total value. Thus the original sponsor could act as a monopolist with respect to the value of the platform as well as the value the sponsor did not create. This is an issue for the sponsor to manage since platform supply- and demand-side users will both reason forward in time and will reduce consumption and participation if the risk of hold-up is too high.

Proposed solution: To encourage *ex ante* investment (i.e. before becoming sunk costs) by developers and affiliation by demand side users, the platform sponsor can contractually commit to forgoing real dollar price hikes on the version of common code acquired by developers. This provides some assurance that on expiration of a developer's proprietary period, the forward value created by the developer will not simply be expropriated by the sponsor. The sponsor, however, need not commit to price levels on its own future development. This leaves the sponsor free to continue adding value through innovation but, analogous to secondary markets in durable goods, future prices will be conditioned by the presence of an inferior substitute. Prices on sponsor enhancements will be proportional to the marginal value created rather than the growing stock of value created, which is socially more efficient.

■ Contributions

The main contribution of this article is to seek resolution of six issues associated with the tension between market incentives that promote innovation against access freedoms that promote social welfare. In essence, a platform sponsor faces an innovation versus access tradeoff seen as a choice between fostering platform adoption and complementary investment versus capturing immediate profits on the platform itself. We show how such a sponsor benefits by using open licensing and property rights to grow an interest in downstream innovation. Opening a portion of a platform's code base can increases the rate of complementary third party investment. Closing downstream innovations, if only briefly, restores incentives to develop, as recently demonstrated by the high rates of developer participation in the iTunes/iPhone application store. Thus, there exist reasons for proprietary licenses to become more open and for free/open licenses to become more closed. Such hybrid models resolve problems of incentives, the right to fork, the right to merge, best of breed hold-up, and monopsony.

A key benefit of decentralized open licensing is the offer of a default contract without negotiation costs. Under traditional closed licenses, a third party developer with a good idea must negotiate access to platform source material. Either through negotiation or by observing the identity of the developer, however, the platform sponsor might discern the idea and appropriate it. In practice, large firms have been accused of this by smaller firms (JACKSON, 1999). This risk reduces their willingness to invest or disclose the idea *ex ante*. In contrast, a free/open license with third party rights to redistribute requires no negotiation and avoids monopsony hold-up from the platform.

In answer to the question of how to balance market incentives and access freedoms, we offer two clarifications. In terms of openness, choose a point such that the revenues lost on the free/open portion balance the net present value of the innovation revenue. In terms of time, the socially optimal protection is not arbitrarily long but neither is it arbitrarily short. Choose a time to bundle such that compensation for downstream developers' investments and opportunity costs balances the need to incorporate best of breed components into a common standard. This stands in contrast to important contributions in economics and law that argue for arbitrarily long and narrow or arbitrarily short and broad (GILBERT & SHAPIRO, 1990; KLEMPERER, 1990) or for time extensions based on investment and

maintenance (LANDES & POSNER, 2003; WAGNER, 2003). The difference is explicit consideration of later period innovation where prices limit reuse and decrease network effects.

In terms of governance, we find that platform sponsors need longer term tenure than downstream developers expressly in order to enforce shorter terms until subsequent disclosure. This makes it possible to "stand on the shoulders of giants." We identify a prisoner's dilemma in which profit motivated developers individually prefer to withhold their innovations from the open platform, but collectively prefer the others contribute. Disclosure and bundling, enforced by the sponsor, resolves problems of multi-party hold-up and the "tragedy of the anticommons."

Finally, a useful contribution of our analysis is to generalize several different types of contracts. This permits comparison of licensing archetypes, ranging from fully open to fully closed, such as those based on GPL, BSD, APIs, and EULAs. Market participants can then ask which contract optimizes a given welfare criterion. The debate on how and when to open a platform is subtle and complex, requiring careful analysis of competing tradeoffs. This research articulates a balance of incentives and openness that promote both the genesis of feature rich new software offset by widespread distribution and network externality benefits from open access.

References

AGHION, P., N. BLOOM, R. BLUNDELL, R. GRIFFITH & P. HOWITT (2005): "Competition and Innovation: An Inverted-U Relationship", *Quarterly Journal of Economics*, May, pp. 701-728.

ARROW, K.J. (1962): "Economic welfare and the allocation of resources for innovation", in R.R. NELSON (Ed.), *The Rate and Direction of Inventive Activity. Economic and Social Factors*, Princeton University Press, Princeton, NJ, 1962, pp. 609–625.

BENKLER, Yochai (2002): "Coase's Penguin, or, Linux and the Nature of the Firm", *The Yale Law Journal*, 112(3).

BESSEN, J. & E. MASKIN. (forthcoming): "Sequential Innovation, Patents, and Imitation", *Rand Journal of Economics*.

BOUDREAU, K. (2008): "Opening the Platform *vs*. Opening the Complementary Good? The Effect on Product Innovation in Handheld Computing". http://ssrn.com/abstract=1251167.

BOUDREAU, K. & A. HAGIU (2009): "Platform Rules: Multi-Sided Platforms as Regulators", in GAWER, A. (Ed.), *Platforms, Markets and Innovation*, Northampton, MA, US: Edward Elgar.

CHANG, H.F. (1995): "Patent Scope, Antitrust Policy, and Cumulative Innovation," *Rand Journal of Economics*, 26(1), 34-57.

DAVID, P. (2004): "Can 'Open Science' be Protected from the Evolving Scheme of IPR Protections?", *Journal of Institutional and Theoretical Economics*, 160(1), p. 9.

DIXIT, A. & J. STIGLITZ (1977): "Monopolistic Competition and Optimum Product Diversity", *The American Economic Review*, 67 (3), 297-308.

EISENMANN, T, G. PARKER & M. VAN ALSTYNE (2006): "Strategies for Two Sided Markets", *Harvard Business Review*, 84(10), 92-101.

EISENMANN, T, G. PARKER & M. VAN ALSTYNE (2009): "Opening Platforms: How, When & Why", in GAWER, A. (Ed.), *Platforms, Markets and Innovation*, Northampton, MA, US: Edward Elgar.

ELZINGA, K., D. EVANS, A. NICHOLS (2001): "United States. *vs*. Microsoft: Remedy or Maladay?", *George Mason Law Review*, 9(3) 633.

FARRELL, J., MONROE, H. & SALONER, G. (1998): "The Vertical Organization of Industry: Systems Competition *vs* Component Competition", *Journal of Economics & Management Strategy*, 7(2) pp. 143-82.

FAUCHART, E., E. VON HIPPEL (2008): "Norms-Based Intellectual Property Systems: The Case of French Chefs", *Organization Science*, 19(2), pp. 187-201.

FINE, C. (1998): *Clockspeed: Winning Industry Control in the Age of Temporary Advantage*, New York: Basic Books.

GAWER, A. (2009): "Platform Dynamics and Strategies: From Products to Services", in GAWER, A. (Ed.), *Platforms, Markets and Innovation*, Northampton, MA, US: Edward Elgar.

GAWER, A. & M. CUSUMANO:

- (2002): *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*, Boston: Harvard Business School Press.
- (2008): "How Companies Become Platform Leaders", *Sloan Management Review*, Winter 2008, 28-35.

GAWER, A & HENDERSON, R, (2007): "Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel", *Journal of Economics and Management Strategy*, Vol. 16, pp. 1–34.

GILBERT, R. & C. SHAPIRO (1990): "Optimal Patent Length and Breadth", *Rand Journal of Economics*, 21(1) 106-112.

GREEN, J.R. & S. SCOTCHMER (1995): "On the Division of Profit in Sequential Innovation", *Rand Journal of Economics*, 26(1), 20-33.

HELLER, M. & R. EISENBERG (1998): "Can Patents Deter Innovation? The Anticommons in Biomedical Research", *Science*, 280(5364) pp. 698-701.

JACKSON, T.P. (1999): "U.S. *v*. Microsoft: Findings of Fact", United States District Court for the District of Columbia.

KLEMPERER, P. (1990): "How broad should the scope of patent protection be?", *Rand Journal of Economics*, 21(1), 113-130.

LAKHANI, K. & E. VON HIPPEL (2003): "How Open Source Software Works: 'Free' User-To-User Assistance", *Research Policy*, 32(6), pp. 923-943.

LANDES, W.M. & R.A. POSNER (2003) "Indefinitely Renewable Copyright", *University of Chicago Law Review*, 70(2), 471-518.

LERNER, J. & J. TIROLE (2002): "Some Simple Economics of Open Source", *Journal of Industrial Economics*, 52, 197-234.

MERGES, R. (2008): "IP Rights and Technological Platforms", Mimeo: UC Berkeley Center for Law and Technology.

MOTTA, M. (2004): *Competition Policy: Theory and Practice*, Cambridge: Cambridge University Press.

NALEBUFF, B. (2004): "Bundling as an entry deterrent", *Quarterly Journal of Economics*, 119, 159-187.

PARKER, G. & M. VAN ALSTYNE:

- (2005): "Two Sided Network Effects – A Theory of Information Product Design", *Management Science*, 51(10), pp. 1494-1504.

- (2008): "Innovation, Openness & Platform Control", Mimeo: Massachusetts Institute of Technology. SSRN.com

RAYMOND, E.S. (1999): *The Cathedral and the Bazaar*, O'Reilly, ISBN 1-56592-724-9.

SALOP, S. (1977): "The noisy monopolist: Imperfect information, price dispersion and price discrimination", *Review of Economic Studies*, 44(3), 393-406.

SHAPIRO, C. & H. VARIAN (1999): "The Art of Standards Wars", Vol. 41, no 2, 8-32.

SPENGLER, J. (1950): "Vertical integration and antitrust policy", *Journal of Political Economy*, 58, 347-352.

STALLMAN, R. (1992): "Why Software Should be Free", Mimeo.

VON HIPPEL, E. & G. VON KROGH (2003): "Open Source Software and the "Private-Collective" Innovation Model", *Organization Science*, 14(2), pp. 209-223.

VALLOPPILLIL, V. (1998): "Open Source Software: A (New ?) Development Methodology". http://catb.org/~esr/halloween/halloween1.html.

WAGNER, R.P. (2003): "Information Wants to Be Free: Intellectual Property and the Mythologies of Control", *Columbia Law Review*, 103(4), 995-1034.

WEST, J.:

- (2003): "How open is open enough? Melding proprietary and open source platform strategies", *Research Policy*, 32, 1259-1285.

- (2006) "The Economic Realities of Open Standards: Black, White and Many Shades of Gray", in S. Greenstein & V. Stango (Eds), *Standards and Public Policy*, Cambridge: Cambridge University Press.

.