

**The Host's Dilemma: Strategic Forfeiture in Platform
Markets for Informational Goods
(forthcoming in HARVARD L. REV. (2010))**

Jonathan M. Barnett

**USC Center in Law, Economics and Organization
Research Paper No. C10-15
USC Legal Studies Research Paper No. 10-18**



**CENTER IN LAW, ECONOMICS
AND ORGANIZATION
RESEARCH PAPER SERIES and LEGAL
STUDIES RESEARCH PAPER SERIES**

University of Southern California Law School
Los Angeles, CA 90089-0071

**The Host's Dilemma:
Strategic Forfeiture in Platform Markets for Informational Goods**

Jonathan M. Barnett¹

Forthcoming, Harvard Law Review

ABSTRACT

Voluntary forfeiture of intellectual assets—often, exceptionally valuable assets—is surprisingly widespread in information technology markets. A simple economic rationale can account for these practices. By giving away access to core technologies, a platform holder commits against expropriating (and thereby induces) user investments that support platform value. To generate revenues that cover development and maintenance costs, the platform holder must regulate access to other goods and services within the total consumption bundle. The tradeoff between forfeiting access (to induce adoption) and regulating access (to recover costs) anticipates the substantial convergence of open and closed innovation models. Organizational patterns in the software and operating system markets are consistent with this hypothesis: open and closed structures substantially converge across a broad range of historical and contemporary settings and commercial and noncommercial environments. In particular, I show that (i) proprietary firms have formed nonprofit consortia and adopted open licensing strategies in order to develop and promote operating systems for the smartphone market, and (ii) leading “open source” software projects are now primarily funded and substantially governed and staffed by corporate sponsors.

¹ Associate Professor, University of Southern California, School of Law; Visiting Professor, New York University, School of Law. I am grateful for comments from participants at the Intellectual Property Scholars Conference at the University of California at Berkeley, School of Law, in August 2010, and the Law and Innovation Symposium at the University of Southern California, School of Law, in March 2010. Research assistance was provided by Jinmin Chen, Daniel Fullerton, Blake Horn, Kawon Lee and Ingrid Newqist. This project was supported by a grant from the Southern California Innovation Project, which is funded by the Ewing Marion Kauffman Foundation.

In June 2008, Nokia paid \$410 million to buy out all other ownership interests in the Symbian operating system², which was then and still is the most widely-used operating system in smartphone devices worldwide.³ That would be a fairly mundane corporate acquisition if it were not for the fact that Nokia immediately transferred its interests in the operating system to a newly-formed nonprofit entity, the Symbian Foundation. That too might be construed as a large but unexceptional act of corporate largesse if it were not the fact that the Foundation is governed by representatives from Nokia together with telecommunications providers, handset makers and other firms that compete with it. To cap off what *is* an exceptional sequence of events, the Foundation then spent two years clearing all third-party rights in the Symbian source code⁴, which, in February 2010, it made publicly available without charge under an “open source” license. Even *more* surprisingly, however, this exceptional giveaway ultimately turns out to be fairly *unexceptional*. From the inception of the information and communication technology (“ICT”) industry, some of the most dominant firms have regularly ceded—that is, given away or distributed at nominal or below-market fees—valuable innovations to all interested parties, including customers and rivals. Examples include some of the industry’s most valuable innovations: to name just a few, AT&T’s forfeiture of the transistor technology in the 1950s⁵, Xerox’s forfeiture of Ethernet local area network technology in 1979⁶, and Intel’s release of the Universal Serial Bus (USB) standard in 1995.⁷ Some of the most fundamental building blocks of the digital economy have been

² See Paul McDougall & J. Nicholas Hoover, *Nokia’s Symbian Deal Rewrites the Smartphone Rules*, INFORMATIONWEEK, June 24, 2008, avail. at http://www.informationweek.com/news/software/open_source/showArticle.jhtml?articleID=208801196.

³ On specific market share, see *infra* Fig. III. While there is no standard industry definition, a “smartphone” can be understood to refer to a mobile phone with advanced capabilities (such as email and internet access) that resemble some of the functions of a personal computer.

⁴ Source code is the human-readable form of the binary code for a software program. Binary code can not be reverse engineered except at great cost, time and labor.

⁵ See Richard C. Levin, *The Semiconductor Industry*, in GOVERNMENT AND TECHNICAL PROGRESS: A CROSS-INDUSTRY ANALYSIS (Richard R. Nelson, ed., 1982).

⁶ See MICHAEL HILTZIK, DEALERS OF LIGHTNING: XEROX PARC AND THE DAWN OF THE COMPUTER AGE 363-64 (1999); John Markoff, *Long Before Microsoft’s Internet War: A Peaceful Ethernet*, N.Y. TIMES, May 18, 1998.

⁷ See Annabelle Gawer & Rebecca Henderson, *Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel*, 16 J. ECON. & MGMT. STRATEGY 1 (2007).

developed at great cost by dominant firms, who then gave away or distributed those innovations at a nominal or below-market fee, often accompanied by complementary support services and tools.

The exceptional incidence and magnitude of giveaway practices in certain ICT markets challenges conventional assumptions that firms will always elect to exert maximal control over intellectual assets, subject solely to enforcement costs. Even—or rather, especially—the most dominant firms’ self-interest will often compel downward adjustments from the level of control that is available as a matter of law or technology. Even more remarkably, this self-interested rationale most strongly recommends forfeiture in the case of the most valuable intellectual assets. In this Article, I identify an incentive design problem that accounts for the voluntary forfeiture of infrastructural categories of technological assets by (ostensibly) commercial and noncommercial entities in the ICT industry. Forfeiture of “crown jewel” technologies is a preferred strategy whenever inducing widespread adoption independently or by contract with third parties is more costly—which, I argue, is a typical case given certain industry-specific characteristics. Competitive pressures force “tough guys” to “play nice”: the market rewards firms and other entities that act generously toward rivals and customers and punishes—often severely—firms and entities that do otherwise.

The key to understanding forfeiture as a rational and typical practice lies in the observation that it appears to be especially common in markets where intermediaries provide a platform technology that matches suppliers of informational inputs with consumers of a resulting bundle of production outputs. Platform markets are characterized by network effects: that is, the platform’s value is an increasing function of the number of users and/or uses. Network effects imply switching costs, which, as compounded by learning costs, implies that users are subject to lock-in effects once the platform has achieved scale. At that point the intermediary (whom I will call the “host”) appears to enjoy pricing power over users. But that commonly expressed view can only be true from a static viewpoint. So long as users anticipate lock-in effects, the host can not induce the user investments that are required for the platform to achieve scale. Hence the host’s dilemma: it must commit to users that the platform will achieve scale *and* that it will not expropriate user investments once scale has been achieved.

This double commitment problem yields forfeiture as both a typical and rational strategy. First, as has been widely observed, if the host *initially* gives away access, it assumes some or all of the risk that the platform will not achieve scale and thereby encourages user adoption. Second, as will be examined in detail, if the host adopts some mix of contractual, organizational and ideological instruments that constrain its ability *subsequently* to regulate access to the platform, it credibly commits against future hold-up. Most dramatically, the host can build the platform and then give it away. This is equivalent to a fail-safe promise against coercive renegotiation of the terms governing platform access. Remarkably, this extreme action is typically adopted in ICT markets.

But the forfeiture solution to the host's dilemma is fatally incomplete. The reason is obvious: it generates no revenues by which the host can cover its platform development and maintenance costs. Hence a perfect solution to underinvestment by users implies underinvestment by the host. Any forfeiture solution must therefore be coupled with a financing solution. That requires regulating access over some portion of the consumption bundle constituted by the platform and complementary goods and services. This inherent tradeoff between forfeiting and controlling access yields an organizational hypothesis. Namely: host entities will tend to implement hybrid structures that reflect a mix of open-access elements (to promote platform adoption) and closed-access elements (to recover costs).⁸ The greater the control forfeited by the host, the stronger its ability to induce user adoption, but the weaker its ability to capture revenues that at least cover development

⁸ In a related line of inquiry, management scholars have recently applied an “openness v. appropriability” tradeoff to understand the extent to which platform holders sometimes open up access in order to provide assurance against user lock-in. See Joel West, *How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies*, 32 RESEARCH POLICY 1259 (2003). For further discussion, see Thomas R. Eisenmann et al., *Opening Platforms: How, When and Why?*, in PLATFORMS, MARKETS AND INNOVATION (Annabelle Gawer ed. 2008), and empirical applications, see Kevin Boudreau, *Opening the Platform v. Opening the Complementary Good? The Effect on Product Innovation in Handheld Computing* (HEC Working Paper 2008) (with application to handheld computing) and Gawer & Henderson, *supra* note 7 (with application to Intel's Architecture Lab). For a related discussion on scientific publications by technology firms as a commitment device, see Oren Bar-Gill & Gideon Parchomovsky, *The Value of Giving Away Secrets*, 89 VA. L. REV. 1857 (2003), which I discuss further subsequently, see *infra* note [30]. For earlier and more theoretical contributions in the economic literature on commitment devices to protect against user lock-in, see *infra* note [52]. This paper extends this line of inquiry in three principal respects: it shows (i) how commitment concerns anticipate the widespread use of forfeiture practices across platform holders, irrespective of commercial or noncommercial motivations; (ii) how firms strategically use non-profit organizational entities, relaxed licenses and “community” norms in order to address users' commitment concerns in operating systems and other infrastructural software markets (in particular, smartphone and open source software markets); and (iii) how the strategic purposes of forfeiture practices cast doubt on the standard normative preference for open innovation models.

costs; conversely, the lesser the control forfeited by the host, the weaker its ability to induce user adoption, but the stronger its ability to capture revenues that at least cover development costs. These parameters substantially constrain the feasible range of organizational choices. The market is unlikely to tolerate entirely closed or entirely open structures: the former due to limited user adoption, in which case the platform can not sustain value, and the latter due to limited revenue accrual, in which case the platform can not cover costs. Put differently: the market rewards generosity so long as it is not excessive—which is to say, so long as it is self-interested.

I apply this theoretical framework to a broad range of historical and contemporary ICT markets, which yields striking results that depart from conventional understandings of the extent to which firms in these markets seek to exercise control over technological assets. This historically-informed inquiry exposes a remarkable commonality of organizational structures across a broad range of ICT technologies in both commercial and noncommercial environments—a result that suggests that market participants are responding to a common economic problem that cuts across otherwise starkly different settings. I start by reviewing the organizational forms used historically to develop and distribute operating systems for personal and enterprise (that is, business) computing devices. Consistent with theoretical expectations, these markets tend to rely on hybrid semi-closed, semi-open structures in order to induce platform adoption within the constraints of business prudence. I then study in greater detail the organizational forms used in two contemporary operating system and software markets. First, I examine the rapidly-evolving smartphone market, where (as illustrated by Nokia’s action) the most widely-used operating systems are now controlled, and made available at no cost, by nonprofit consortia comprising most of the world’s leading handset makers, telecommunications providers, and semiconductor chip makers. These firms employ non-profit organizational structures, open source licenses and “community” norms as strategic tools by which to elicit developer investments that are critical to securing platform adoption. Second, I show that a similar structure characterizes leading open source software projects that are ostensibly organized for noncommercial purposes.⁹ Leading open source projects are governed by hierarchically-organized nonprofit

⁹ For a definition, see *infra* note [11] and accompanying text.

foundations that are substantially managed and staffed by commercial sponsors, who provide virtually all funding and most code contributions. These findings, which rely on a detailed examination of organizational structures, contributor affiliation, and funding information, depart sharply from the conventional characterization of open source projects in the legal (and some of the economic) literature as spontaneously-organized communities of intrinsically-motivated programmers who contribute code without compensation.¹⁰

In both the smartphone and open source software markets, controlled generosity follows from economic self-interest: implicit or explicit consortia of commercial firms seek to commit against host opportunism, which is designed to induce adoption of a platform technology that promotes those firms' sale of complementary goods and services. This commonality of organizational structure across ICT markets casts doubt on any meaningful distinction between open and closed innovation structures. For-profit firms adopt open structures in order to commit against host opportunism while (ostensibly) not-for-profit communities adopt closed structures in order to enable the recovery of development and maintenance costs and avoid platform demise. That descriptive ambiguity in turn casts doubt on the normative presumption that policy interests inherently favor the adoption of open over closed structures. If open and closed structures (and all intermediate variants) simply reflect strategic approaches to the underlying tradeoff between controlling host opportunism and enabling cost recovery, then the choice of organizational form would appear to be a matter of social indifference that provides no basis for government intervention to guide market outcomes. Access policies, as implemented through some mix of closed and open organizational components, are simply part of the consumption bundle offered by competing providers in the strategic pursuit of market share. Moreover, there is no assurance that open structures even promote consumer welfare. That is because forfeiture inherently exerts entry-deterrent effects that can protect dominant firms against potential competitors. A host who forfeits its platform technology compels stand-alone platform providers to exit

¹⁰ See, e.g., Yochai Benkler, *Coase's Penguin, or, Linux, and the Nature of the Firm*, 112 YALE L.J. 369 (2002); James Boyle, *The Second Enclosure Movement and the Construction of the Public Domain*, 66 J. LAW & CONTEMPORARY PROBLEMS 33 (2003). These works build upon views expressed by the most well-known popular work on the subject, see ERIC S. RAYMOND, *THE CATHEDRAL AND THE BAZAAR: MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY* (1999).

the market, which may then enable the host to extract rents through complementary markets in which it has a competitive advantage. Whether those reallocations of industry rents within the total consumption bundle leave end-users in a superior, inferior or indifferent position is ambiguous in general and may be difficult to answer in any particular case.

Organization is as follows. In Part I, I describe illustrative forfeiture practices in ICT markets. In Part II, I describe the host's dilemma and possible solutions through contract, integration and forfeiture. In Part III, I show how ICT firms have addressed the host's dilemma through a mix of open and closed access policies that govern operating systems for the personal, enterprise, and mobile computing markets. In Part IV, I discuss how the strategic motivations behind forfeiture practices complicate policy preferences for open over closed innovation models.

I. Voluntary Forfeiture: A Typical Practice

Loosely defined, open source software ("OSS") is an industry segment where software products and the underlying source code are released at no fee subject to relaxed contractual restrictions on use and distribution.¹¹ Contrary to popular (and some scholarly) characterizations as a novel form for technological innovation, OSS (which dates roughly from the early 1990s) is merely a recent addition to a larger and long-standing set of forfeiture practices in ICT markets. Some examples are familiar to end-users who stand at the end of the ICT supply chain: Google provides access to its search engine at no cost; Adobe releases basic versions of its Reader software at no cost; Microsoft releases its browser for download at no cost; and so on. Other examples are even more fundamental and familiar to professional users that stand at intermediate points of the supply chain. I include a selection of some of the most dramatic forfeitures in the Table below.

¹¹ By contrast, proprietary software is released in non-human-readable object-code form for a fee and under strict contractual restrictions on use and distribution.

Table I: Selected Forfeiture Actions in ICT Markets¹²

Date	Firm	Forfeiture Action
1940s-50s	Bell Labs (AT&T)	Licensed transistor and related technologies at nominal royalty, subject to cross-licensing obligation. Licensing practices formalized by 1956 consent decree.
1979	Xerox	Licensed Ethernet local area network technology at nominal fee. In 1983, it contributed all Ethernet patents to the nonprofit Institute of Electronics and Electrical Engineers.
1985-present	Microsoft	Disclosed some Windows application programming interfaces to independent software developers.
1995	Intel	Released Universal Serial Bus interface technology through USB Implementers Forum (USB-IF), a nonprofit trade organization.
1999	Netscape	Released source code for its browser technology
1999	IBM	Announced \$1 billion commitment to (open source) Linux operating system
2002	IBM	Released source code to software development tool platform (acquired for \$40 million); subsequently donated code to nonprofit Eclipse Foundation.
2005	Sun Microsystems	Released portions of source code for Solaris operating system; later transferred governance to nonprofit community foundation.
2006-07	Sun Microsystems	Released source code for Java programming language, subject to limited contractual restrictions.

As the Table shows, recent patterns of corporate generosity are not a contemporary fashion; rather, this is consistent with historical behavior in the ICT industry from its inception. Contrary to natural intuitions, some of the most dominant firms have regularly

¹² Sources as follows. On Bell Labs, see immediately following discussion; on Xerox, see HILTZIK, *supra* note __, at 363-64, Markoff, *supra* note __; on Microsoft, see immediately following discussion; on Intel, see Gawer & Henderson, *supra* note __ and INTEL, UNIVERSAL SERIAL BUS, avail. at <http://www.intel.com/technology/usb/index.htm>; on Netscape, see Jacques Bughin et al., *The Next Step in Open Innovation*, MCKINSEY QUARTERLY (2008), available at https://www.mckinseyquarterly.com/next_step_in_open_innovation_2155; on IBM's investment in Linux and other related actions, see *infra* notes __ and accompanying text; on IBM's donation to Eclipse, see Martin LaMonica, *Eclipse to split from IBM*, CNET NEWS, Jan. 20, 2004, http://news.cnet.com/Eclipse-to-split-from-IBM/2100-7344_3-5143421.html; Robert McMillan, *Will Big Blue eclipse the Java tools market?*, JAVAWORLD.COM, Feb. 22, 2002, <http://www.javaworld.com/javaworld/jw-02-2002/jw-0222-eclipse.html>; Darryl K. Taft, *IBM Donates Development Assets to Eclipse*, EWEEK.COM, Oct. 12, 2005, <http://www.eweek.com/c/a/Application-Development/IBM-Donates-Development-Assets-to-Eclipse/>; on Sun's actions, see Raghu Garud & Arun Kumaraswamy, *Changing Competitive Dynamics in Network Industries: An Exploration of Sun Microsystems' Open Systems Strategy*, 14 STRATEGIC MGMT. J. 351 (1993).

given away some of their most valuable technologies—including, by implication, to actual and potential rivals and customers. Below I discuss in further detail two of the most notable illustrations.

A. Bell Labs: Open Licensing

The licensing practices of Bell Labs, the research laboratory founded by AT&T in 1925 (and owned since 2001 by Lucent Technologies, a spin-off entity), probably constitute the single greatest act of corporate generosity. Bell Labs is credited with approximately 40,000 inventions, including (among other things) the transistor, the UNIX operating system, and key technologies behind cellular mobile communications.¹³ AT&T made these technologies available at nominal royalties subject to a cross-licensing obligation. While these policies were mandated under a 1956 consent decree settling federal antitrust litigation, AT&T had instituted roughly the same policies prior to the consent decree and exceeded those policies thereafter.¹⁴ Roughly from the inception in the 1940s of serious research by Bell Labs on the transistor (the basis for virtually all modern ICT technologies), AT&T had adopted a policy to actively disseminate the technology and even assist third parties in using it.¹⁵ A former AT&T executive states so explicitly: “Bell Labs’ first important policy was not to keep transistor information secret. Not only was it not kept a secret, but we actively expounded the art as well as the science of practicing the technology. Several seminars were held in the 1950s when we effectively told all we knew about transistor technology.”¹⁶ Remarkably, this statement tracks almost exactly an Intel executive’s statement concerning his firm’s open licensing of the Universal Serial Bus standard several decades later¹⁷: “We developed the [USB]

¹³ See NATHAN M. MULLER, DESKTOP ENCYCLOPEDIA OF TELECOMMUNICATIONS (2002).

¹⁴ See Levin, *supra* note 5; Nathan Rosenberg, *Endogenous Forces in Twentieth-Century American*, in ENTREPRENEURSHIP, INNOVATION, AND THE GROWTH MECHANISM OF THE FREE-ENTERPRISE ECONOMICS 80 (Eytan Sheshinski et al. eds., 2007).

¹⁵ See J.E. TILTON, INTERNATIONAL DIFFUSION OF TECHNOLOGY: THE CASE OF SEMICONDUCTORS (1971).

¹⁶ See Morgan Sparks, *Morgan Sparks Reflects on 25 Years of Transistors* 343-44, BELL LABORATORIES RECORDS (1972).

¹⁷ As most readers are probably aware, the USB standard is a peripheral interface that enables communication between a computer and external devices such as printers, keyboards, flash memory disks, and a variety of other items.

code and . . . we also made it available to anybody in the industry.”¹⁸ As I will show, a common logic explains the broad persistence of this practice in technology markets.

B. Microsoft: Application Programming Interfaces

It is sometimes overlooked that Microsoft is one of the historical leaders in the open development of software. As is widely observed, Microsoft’s success rests in part on its release of Windows APIs to outside developers¹⁹ and extensive efforts to construct a product architecture and communications infrastructure that facilitates third-party development of complementary applications.²⁰ For access to application protocol interfaces (“APIs”)²¹ and related technical information, Microsoft has often assessed a zero or even negative fee taking into account the software development tool kits and support services it provides.²² Moreover, Microsoft has incurred significant costs—both direct programming costs and indirect costs in the form of product quality²³—in order to make its APIs “backward compatible” across Windows versions²⁴, which allows existing applications to operate on newer versions of Windows. This is not to say that Microsoft does not restrain use of other parts of its technology. It is simply to observe that Microsoft has given away access to technological assets over which it could feasibly and

¹⁸ See Gawer & Henderson, *supra* note 7.

¹⁹ See Shane Greenstein, *Open Platform Development and the Commercial Internet*, in PLATFORMS, INNOVATION AND COMPETITION (Annabelle Gawer ed., 2009); Renata B. Hesse, *Section 2 Remedies and U.S. v. Microsoft: What Is To Be Learned?*, 75 ANTITRUST L.J. 847 (2009); MARCO IANSITI & ROY LEVIEN, THE KEYSTONE ADVANTAGE: WHAT THE NEW DYNAMICS OF BUSINESS ECOSYSTEMS MEAN FOR STRATEGY, INNOVATION, AND SUSTAINABILITY (2004); William H. Page, *Mandatory Contracting Remedies in the American and European Microsoft Cases*, 75 ANTITRUST L.J. 787 (2009).

²⁰ See Greenstein, *supra* note 19; IANSITI & LEVIEN, *supra* note 19; MICHAEL CUSUMANO & RICHARD W. SELBY, MICROSOFT SECRETS: HOW THE WORLD’S MOST POWERFUL SOFTWARE COMPANY CREATES TECHNOLOGY, SHAPES MARKETS, AND MANAGES PEOPLE (1995). Tellingly, Intel—Microsoft’s partner in crime from the perspective of certain antitrust authorities—has pursued the same cooperative strategy with respect to some of its most fundamental technologies. See Gawer & Henderson, *supra* note 7.

²¹ An API is a language and message format used by an application program to communicate with the operating system or other application program

²² See CUSUMANO & SELBY, *supra* note 20; DAVID S. EVANS ET AL., INVISIBLE ENGINES: HOW SOFTWARE PLATFORMS DRIVE INNOVATION AND TRANSFORM INDUSTRIES 66 (2006).

²³ Backward compatibility can reduce product quality to the extent that it limits freedom of development in newer versions.

²⁴ See Greenstein, *supra* note 19; IANSITI & LEVIEN, *supra* note 19.

legally have done otherwise.²⁵ The fact that its less successful competitors (unwisely) chose *not* to do so—Apple, who pursued a largely closed development strategy in competing with Windows, and, at an earlier stage, European competitors who persisted in producing software for proprietary hardware architectures²⁶ — illustrates this point nicely.

II. Voluntary Forfeiture: A Rational Practice

There is now a puzzle to be explained. Why do economically rational actors give away valuable—sometimes exceptionally valuable—technological assets? Non-economic factors such as altruism, ideology or intrinsic interest, which are sometimes proposed to explain individual contributions of knowledge assets in ostensibly non-commercial settings such as OSS²⁷, are localized accounts that do not extend to commercial entities obligated to maximize profits. Existing non-altruistic explanations include high enforcement costs, a strategy to preempt patenting by other firms²⁸, a strategy to recruit researchers who wish to accumulate reputational capital in the scientific community, or a strategy to build a large installed base.²⁹ These explanations too have merit but are generally *ad hoc* case-specific explanations or, in the last case, are incomplete for reasons described below. In this Section, I propose a general but simple rationale: host entities forfeit platform assets in order to commit credibly against

²⁵ More recently, Microsoft has undertaken further commitments to expose its technology (including portions of the Windows source code) to outside parties, including the Shared Source Initiative (2005), the Open Specification Promise (2006) and the Interoperability Principles (2008). I omit these more recent actions as it is difficult to ascertain the extent to which Microsoft undertook these initiatives as a preemptive tactic to deter further governmental or private antitrust prosecution.

²⁶ See David Lancashire, *Code, Culture and Cash: The Fading Altruism of Open Source Development*, FIRST MONDAY (SPECIAL ISSUE NO. 2) (Oct. 3, 2005), available at <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/1488/1403>.

²⁷ See *supra* note [10].

²⁸ See Robert P. Merges, *A New Dynamism in the Public Domain*, 71 U. CHI. L. REV. 183 (2004); Gideon Parchomovsky, *Publish or Perish*, 98 MICH. L. REV. 926 (2000); Doug Lichtman et al., *Strategic Disclosure in the Patent System*, 53 VAND. L. REV. 2175 (2000).

²⁹ See Joseph Farrell & Garth Saloner, *Installed Base and Compatibility: Innovation, Product Preannouncements and Predation*, 76 AM. ECON. REV. 940 (1986); Lisa N. Takeyama, *The Welfare Implications of Unauthorized Reproduction of Intellectual Property in the Presence of Demand Network Externalities*, 42 J. INDUSTR. ECON. 155 (1994).

expropriating users' investments in those assets.³⁰ Even more simply: forfeiture signals good faith.

A. Platform Markets

Any ICT platform—which may be constituted by hardware, software or an operating system³¹—must enable users to transact at a lower cost relative to transacting directly (or through the next-best platform technology); otherwise, it will not be adopted.

Transaction-cost savings plus associated trading gains are amplified by network effects. That is: any user's gains are an increasing function of the number of other users of, and/or uses for, the platform technology.³² As shown in the Figure below, users consist of end-users (EU) and developer-users (DU), each of which is connected by three possible transaction paths that run through the platform: DU-EU; EU-EU; and DU-DU.³³ For developer-users, platform value is an increasing function of the number of end-users (on the path EU-EU) and the number of complementary uses developed by other

³⁰ For contributions in the management literature that have explored the extent to which platform owners open up access in order to commit against lock-in, see *supra* note 6. The following discussion provides a somewhat narrower formulation of that argument (in part because I believe that (transiently) dominant firms' lock-in capacities are more limited than is often stated to be the case), which is then applied in the remainder of the Article to selected operating systems markets. In a related context, Profs. Bar-Gill and Parchomovsky have argued that an original innovator will publish technological knowledge that could have been kept secret or patented in order to credibly reserve to follow-on innovators a portion of the surplus generated by a stream of cumulative innovation (and over which it is otherwise difficult to contract). See Oren Bar-Gill & Gideon Parchomovsky, *The Value of Giving Away Secrets*, 89 VA. L. REV. 1857 (2003). My analysis is consistent with (and my empirical evidence further confirms) that argument; however, the controlled forfeiture practices on which I focus are more elaborate than merely abandoning knowledge to the public domain. That is for two reasons: (i) credibly committing against expropriating user investments requires sequestering knowledge in an entity over which the host entity can not exercise control; and (ii) any forfeiture action requires regulating access over some complementary good in order to generate revenues to cover the host's development and other costs.

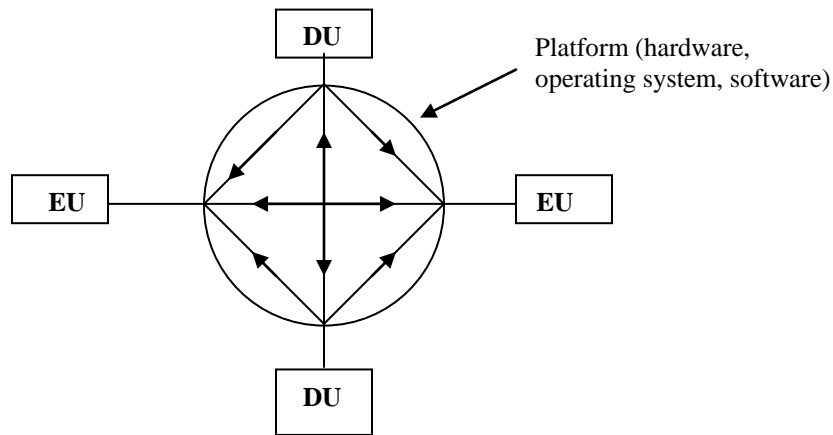
³¹ Slightly more and less expansive definitions of this constituent set are sometimes used (for example, operating system can be construed as a type of software while the software category can be subdivided to include middleware applications). For a fuller explanation in non-technical terms, see MARC H. MEYER & ALVIN P. LEHNERD, *THE POWER OF PRODUCT PLATFORMS* (1997).

³² This corresponds, respectively, to what the economics literature calls direct network effects (related to the number of users) and indirect network effects (related to the number of uses developed by third parties).

³³ For simplicity, the Figure does not reflect that other intermediaries will typically occupy points on the transaction path between end-users and developer-users—in particular, systems integrators (such as original equipment manufacturers in the personal computer market). In general, absent market imperfections, intermediation will occur at any point on the transaction path to the extent it generates net savings over direct transactions at that same point.

developers (on the path EU-DU); for end-users, platform value is an increasing function of the number of other end-users (on the path EU-EU) and the number of complementary uses developed by developers (on the path EU-DU).³⁴ Microsoft Windows, Sony Playstation and the Apple iPhone confer transactional gains by (i) connecting user-developers (e.g., video game developers) and end-users (e.g., video game players) and (ii) connecting end-users to other end-users (e.g., iPhone users) or user-developers to other user-developers (e.g., developers of software applications for Windows who effectively make available an applications suite to consumers).

Figure I: Transaction Paths in ICT Platform Markets



The interdependent demand functions that characterize platform-based markets imply that user adoption rates can exhibit both negative and positive feedback effects. If there are no applications written for Windows, it has virtually no value; if there are no end-users of Windows, its applications have no value; however, as more applications are written for Windows, it increases in value, attracts more end-users, which in turn induces more developers to write applications for Windows, and so on. To succeed, any platform must trigger and maintain positive feedback effects by sustaining adoption by the relevant set of interacting user groups. Failure to do so triggers negative feedback effects that

³⁴ For simplicity, let's assume (i) neither user group suffers from congestion costs, (ii) both users and developers enjoy constant increasing returns from additional developers or users, respectively, and (iii) users and developers value, respectively, all types of developers and users equally (or equivalently, users and developers are homogenous).

erode the platform's value or stunt adoption altogether. That is in turn exacerbated by the fact that platform markets exhibit winner-take-all effects: the transaction-cost savings derived from using a single platform drive users to—and, in a period of decline, *away*—from the same platform. Even the most dominant platform therefore inherently occupies a precarious position: it can be slow to start and can suffer a rapid demise.

This proposition may appear surprising given Microsoft Windows' still-dominant position among operating systems for desktop computing.³⁵ But just a slightly broader view of technology markets shows that even a dominant platform often occupies a fragile position. A few examples suffice. IBM virtually created the personal computer industry in 1981, immediately achieving a 75% market share, but was rapidly overwhelmed by clones and exited from the market entirely in a sale to Lenovo in 1994; Palm distributed the first successful handheld computing device, achieving 70% U.S. market share by 1997³⁶, but was overwhelmed by RIM's Blackberry device in the early 2000s and was sold in a distressed transaction to Hewlett Packard in 2010; Netscape was the prevailing internet browser, achieving an 80% market share in 1995, but lost the leadership position to Microsoft's Internet Explorer by 1999 and had a nominal market share by the early 2000s.³⁷ The best case may be America Online ("AOL"): its apparent dominance of the internet service portal market was so powerful (60% market share as of 1997) that, in 2001, it could merge (effectively, acquire) the media conglomerate, Time Warner; by 2009, the combined entity had spun off the declining AOL, which, as of July 2010,

³⁵ Even Microsoft's dominance in desktop computing is vulnerable to competition in adjacent markets for partially substitutable computing devices. As of May 26, 2010, Apple had surpassed Microsoft in stock market capitalization, reflecting the partial displacement of the personal computer market by the smartphone and tablet devices market (where Apple's products are among the market leaders and Windows-based devices are laggards). See Miguel Heft & Ashlee Vance, *Apple Passes Microsoft as No. 1 in Tech*, N.Y. TIMES, May 26, 2010. For further information on the smartphone market, see *infra* Fig. III and accompanying text.

³⁶ See PALM, PRESS RELEASE, *Dataquest and PC Data Research Studies Place U.S Robotics PalmPilot at Lead of Thriving Handheld Computer Market* (May 12, 1997), avail. at <http://investor.palm.com/releasedetail.cfm?ReleaseID=336348>.

³⁷ As described subsequently, the Netscape code later formed the basis for the open-source Firefox browser, which is managed by the Mozilla Foundation. Reflecting the volatility of platform dominance, Firefox now has 31% market share as of August 2010 (based on median value of multiple estimates of worldwide browser market share based on usage, as set forth at "Usage share of web browsers—Summary Table", avail. at http://en.wikipedia.org/wiki/Usage_share_of_web_browsers#Summary_table) (last visited October 3, 2010).

accounted for a negligible 2.2% of the U.S. “core search” market.³⁸ The appearance of platform dominance can often be illusory—misleading even the most sophisticated academic observers³⁹, antitrust judges, and market investors—and the movement from leader to laggard can often be swift and brutal.

B. Intertemporal Externalities

The most obvious obstacle to eliciting user adoption follows the well-known logic of a collective action problem. Any potential user knows that the platform has no value unless it achieves sufficient adoption from other users. End-users are therefore reluctant to adopt a platform until a large mass of other end-users and/or applications has materialized, which in turn means that developers decline to invest in developing applications for a platform that has not yet been widely adopted, which in turn exacerbates end-users’ unwillingness to adopt the platform, and so on. This “chicken and egg” problem is not without at least a partial remedy. The host can assume the cost and risk of scaling up the platform by providing access to early adopters at a zero or even negative price.⁴⁰ That explains risky gambles such as JVC’s decision in the 1970s to license widely the VHS technology for video cassette recorders, which prevailed over the competing Betamax technology that Sony kept to itself⁴¹, or Xerox’s decision in the 1980s to license its Ethernet local area network technology at a nominal fee, which made it the global networking standard and displaced the then-prevailing IBM technology.⁴²

³⁸ See CBR Staff Writer, *Microsoft, Yahoo! Sites Gain US Search Share, says comScore*, CBR SOFTWARE (Jul. 15, 2010), http://media.cbronline.com/news/microsoft-yahoo-sites-gain-us-search-share-says-comscore_150710.

³⁹ Commentators’ laments of “unbeatable” network monopolies often prove to be short-sighted. Hence, Palm Pilot was once cited as the case of a network good that attracts market adoption through positive feedback effects, which then purportedly erect an entry barrier that is difficult to overcome. See, e.g., Steven C. Salop & Craig R. Romaine, *Preserving Monopoly: Economic Analysis, Legal Standards and Microsoft*, 7 GEO. MASON L. REV. 617, 621 (1999). With hindsight, Palm Pilot demonstrates the ease with which market dominance can be lost. In 2010, HP acquired Palm at a heavily distressed price.

⁴⁰ See Nicholas Economides, *Network Externalities, Complementarities and Invitations to Enter*, 12 EUR. J. POL. ECON. 211 (1996).

⁴¹ See Michael Cusumano et al., *Strategic Maneuvering and Mass-Market Dynamics: The Triumph of VHS over Beta*, 66 BUS. HIST. REV. 51 (1992).

⁴² That Xerox failed to capitalize on its success (which it had hoped would promote the sale of complementary product lines where it had a competitive advantage) strengthens the host’s *second* credible commitment (discussed immediately below): namely, that it will have limited ability to exercise pricing

These costly transfers from the host to early adopters function as a bond posted by the host, who stands to suffer a financial penalty if the platform fails to scale as required. That bonding solution can provide some assistance in eliciting user adoption. However, as will now be discussed, it is incomplete in a fundamental respect.

C. The Host's Dilemma

Even if the host can post a bond by which to persuade users that the platform will achieve sufficient scale, it still has not overcome obstacles to user adoption. That is because the host still can not persuade users that it will leave them with a net positive gain after the platform has achieved scale. The user is wary of false gifts: burning money at an initial stage is an empty signal if it simply enables the host to gain at the expense of users at some subsequent stage. This difficulty gives rise to the host's dilemma.

1. *The Simple Case*

Suppose that user adoption requires making a nonsalvageable investment in learning to use the platform and, more generally, adapting existing activities to it. This is true in the case of end-users and true to a substantially greater extent in the case of developer-users, who must invest substantial sums in developing, marketing and supporting applications for use on the platform. If this is assumed to be the case across platforms, then learning costs imply switching costs equivalent to the costs of learning how to use any other competing platform. Switching costs in turn provide an expropriation opportunity for the host, who will extract value from users equal to the switching costs that users would incur in migrating to the next-best platform technology. The host can do so through various devices, including (among other things) increasing usage fees, reducing technical assistance, making platform modifications that reduce the value of specific investments in the existing platform, or integrating forward into a developer-user's market. All of these are familiar accusations leveled against dominant holders of platform technologies—for example, Microsoft or Intel—in informational goods markets. In the language of transaction cost economics, users make *ex ante* investments that are

power over early adopters, who will be free to purchase from competing providers. See HILTZIK, *supra* note 6, at 363-64.

specific to the platform, which implies an *ex post* expropriation opportunity for the host, who then can regulate access so as to extract value from locked-in users. In the extreme case where no other platform exists in the market and the only remaining alternative is transacting directly, the host will extract from users nearly all the transaction-cost savings and associated gains from use of the platform. This implies a paradoxical result.

Precisely at the point where a platform has achieved the highest levels of user adoption, users derive both the greatest *potential* benefit from the platform (since the users' gains from network effects are highest), but, given the expropriation threat (and assuming the lack of competing platforms), the lowest *actual* benefit (since the users' gains will be almost completely confiscated).

But this statement is intentionally myopic. Assume for the moment that users have perfect foresight. Then, at the initial point at which the host offers access to a new platform, the user will decline—even if access is offered at a zero price. The user anticipates that, after the platform has achieved scale, it will be subject to expropriation by the host. The user may therefore never realize a net positive return *ex post* on its specific investments in the platform, in which case it rationally declines to adopt the platform *ex ante*. Where the user anticipates complete expropriation of its gains by the host, there is no positive price at which the host can offer access to the platform and elicit adoption. That holds true even over a certain range of negative prices equal to the specific investments that the user anticipates the host will expropriate from it. Hence the host's dilemma: unless it can commit against future expropriation *ex post*, the host can not induce platform adoption *ex ante*. The real problem is not that the host will expropriate value from locked-in users; rather, the problem is that the host can not persuade users that it will *not* expropriate value from users after scale has been achieved. The result: the platform is never adopted at all.

2. *The Complex Case*

In defense of the conventional view, it might be objected that the perfect foresight assumption—that is, the fully rational user—is unrealistic. Platforms *are* adopted in ICT markets. Consistent with the argument set forth above, that suggests that users fail to anticipate expropriation opportunities and the host will deceive users into making foolish

platform investments. That concern does not seem reasonable in the case of developer-users, who make substantial investments in the platform, are sophisticated parties, and are subject to external market discipline and, at the managerial level, internal firm discipline to act in conformity with business rationality. But, under certain behavioral assumptions that are sometimes given credence in consumer-goods settings, this scenario may be reasonable in the case of end-users, who may make smaller specific investments in the platform and have weaker incentives to invest resources in exercising perfect foresight.⁴³ Note, however, that, for this objection to hold, it must be the case that users on *both* sides of the market lack foresight (at least in cases where the two user groups are not identical). If the host only misleads end-users, developers will still decline to invest and end-users will observe the lack of applications and decline to adopt given the anticipated absence of network effects. Put simply: only developer-users have to be sophisticated to protect all users against host opportunism.

I will nonetheless grant this objection and implausibly assume for the sake of argument that users on both sides of the market have no ability to anticipate future opportunism by the host. That is: both user groups are myopic—including software developer firms that place substantial capital at stake.⁴⁴ That would allow the host to elicit adoption to the extent that users fail to anticipate its future opportunistic behavior. But this will still make no difference. Unless we further assume (even more implausibly) that users are both perfectly myopic and perfectly forgetful, the host would only have a single opportunity to engage in opportunistic behavior. That is an insufficient solution in the case of any repeat-player host that seeks to maximize long-term profits through repeated adoption of platform extensions. Microsoft can not make any more money by

⁴³ Even that is doubtful: end-users' interests may be effectively represented by original equipment manufacturers and other systems integrators, which effectively purchase components on behalf of end-users. Of course it could be argued that even far-sighted systems integrators will rationally exploit end-user myopia, thereby acting as if they too are myopic.

⁴⁴ In its antitrust allegation that Microsoft had induced developers to write programs for the Windows-specific Java development tool (rather than the cross-platform development tool offered by Sun), the federal government had been forced to rely precisely on the implausible claim that professional developers were unaware of the differences between these products. On this claim (and the lack of supporting evidence), see David McGowan, *Has Java Changed Anything? The Sound and Fury of Innovation Litigation*, 87 MINN. L. REV. 2039 (2002).

selling the Windows operating system again.⁴⁵ It must convince users to buy the latest version of Windows, which explains in part why it has released 16 versions of the Windows operating system for desktop computing from 1985 through the present.

At each release point, the host must re-confront the host's dilemma: that is, it must induce user adoption or risk losing its investment in the latest release. Those investments are substantial to exorbitant in the case of platform technologies. Estimated development costs for a substantial extension to an operating system typically reach several billions of dollars and appear to be escalating: development costs for Windows 2000 were reportedly \$1 billion while development costs for Windows Vista (released in 2007) were reportedly \$6 billion.⁴⁶ Eliciting user adoption to any platform extension, and thereby recovering these substantial investments, rests on maintaining a past record of good behavior. Given the sums at stake, failure to do so exposes the host to substantial or exorbitant financial penalties. The host will therefore have powerful incentives to refrain from exploiting expropriation opportunities even if any current platform release has achieved scale. Even if all users are myopic but not amnesiacs (or just so long as developer-users are not amnesiacs), a repeat-player host has little to no ability to *rationaly* exploit its expropriation opportunity. User lock-in is a virtual impossibility.

It might therefore be concluded that the host's dilemma disappears in the typical repeat-play environment. That would be a happy but somewhat curious result given the widespread impression that dominant holders of platform technologies—again, Microsoft and Intel—do not seem always to behave “nicely” toward existing users. Of course, even a widespread impression may be mistaken (as will be shown, that is precisely the case with respect to the conventional understanding of OSS); but let's assume for the sake of argument that host entities do sometimes expropriate value from existing users. Three contingencies can explain why even fully rational repeat-player host entities would act in this manner—and without relying on implausible or “stretched” assumptions of

⁴⁵ This assumes (as is the case) that Microsoft sells Windows on a one-time basis rather than leasing access to it on a continuous basis.

⁴⁶ All costs are estimates. See Marius Oiaga, *Vista – a \$6 Billion Dollar Operating System*, SOFTPEDIA (2007) (on Windows Vista), available at <http://news.softpedia.com/news/Vista-a-6-Billion-Dollars-Operating-System-44096.shtml>; Operating System Documentation Project, Windows Family Operating Systems, avail. at http://www.operating-system.org/betriebssystem/_english/bs-windows.htm (on Windows 2000).

universally myopic and amnesiac users (or, for that matter, irrational host entities). First, if the host's managers are not infinitely-lived across generations of platform extensions, then managers may have short-term incentives to expropriate users' investments in any given platform extension even if doing so is not consistent with long-term profit maximization. That would mean that platform holders (and users) are victims of platform managers. Second, if host opportunism is punished by anything other than the irrevocable exit assumed in stylized models of repeat-play prisoner's dilemma games, the host may conclude that the short-term gain from expropriation exceeds the long-term reputational penalty.⁴⁷ Third, at the point at which any platform (or platform extension) is released, the host can not commit that it is not playing the final period of a finitely repeated game, in which case the repeat-play incentives to avoid opportunistic behavior are diminished. This corresponds in particular to a declining industry or declining firm that has diminished reputational incentives to avoid exploitative behavior. Anticipating some or all of these contingencies, even mildly sophisticated users will decline to adopt or, at least, underadopt: that is, they assign a positive likelihood that the host may expropriate in the future, in which case users will only adopt subject to a discount that reflects that contingency. Hence the host's dilemma persists: even if it is a repeat player, the host can not fully commit against expropriation risk within these limited but typical contingencies and therefore can not induce users to adopt the platform (or can not do so without offering a discount).

3. Possible Solutions

The host's dilemma implies that platforms will be substantially underadopted. Clearly that is not the case: platforms are endemic in ICT markets. Therefore the analytical task now lies in explaining how platform holders overcome the host's dilemma. Resolving this puzzle will in turn resolve the original puzzle of voluntary forfeiture, which turns out to provide the most potent solution to the host's dilemma. To reach this conclusion, we must assess the relative effectiveness of three devices by which to address the host's commitment problem: *contract*; *integration*; and *forfeiture*. While

⁴⁷ See Michael Kende, *Profitability Under an Open Versus a Closed System*, 7 J. ECON. & MGMT. STRATEGY 307 (1998).

forfeiture is almost certainly the most effective device to promote platform adoption, it has a crucial defect: it does not permit the host to cover its costs.

a. *Contract*

The host can attempt to write a contract that binds it against opportunistic behavior. This is a meaningful but imperfect solution to the host's dilemma for three principal reasons: (i) the user's ability to enforce the contract is limited by the host's life, solvency and legally-attachable assets, (ii) legal action is costly and uncertain (and, given collective-action constraints, likely to be severely underfunded whenever there is a diffuse user population), and (iii) specification costs may make it difficult to address all possible actions by which the host can expropriate value from users. The last point is especially applicable to a platform technology—in IT industry jargon, a “horizontal” application—that may be applied across a broad and difficult-to-anticipate range of uses. It may be difficult to write a contract that can specify all ways in which a host could expropriate users' investments through incomplete efforts at maintaining backwards compatibility or insufficient efforts to provide technical support. Even if we unreasonably assume that all possible expropriation opportunities can be foreseen (or more precisely, can be foreseen at a reasonable cost), it is still likely that definitional limitations will make it difficult to craft language that excludes expropriation opportunities without excluding other legitimate business actions or exposing the host to illegitimate claims. Contract provides some meaningful ability by which to provide assurance to users but it can not entirely, and may not even substantially, eliminate the host's dilemma.

b. *Integration*

Where contracts cannot be written to provide users with sufficient assurance against host opportunism, the host may elect an alternative strategy. As is well-known in the transaction cost literature, the threat of ex post opportunism can be eliminated or mitigated through vertical integration.⁴⁸ For example, the host can employ all developer-

⁴⁸ For the classic source, see Oliver E. Williamson, *The Vertical Integration of Production: Market Failure Considerations*, 61 AM. ECON. REV. 112 (1971).

users, which provides developers with fixed compensation that reduces exposure to host opportunism at the cost of reduced exposure to any project's upside. Or the host can purchase a developer entity that has achieved success in developing an application for use on the platform. If the host makes such acquisitions regularly, then it may be implicitly understood to offer a standing reward for third parties that develop applications that enhance the platform's value (a view that could describe Microsoft, Oracle or Cisco Systems, which regularly make acquisitions as a source of "pre-packaged" R&D). But integration suffers from (at least) four drawbacks. First, while the host can mitigate expropriation risk on the developer side, it can not employ the other side of the market—end-users, without whom most of the transaction paths can not be completed. Second, an employment relationship can not replicate the direct connection between investment and profit that results in the high-powered incentives characteristic of a contractual relationship or that would prevail in the case of third-party developers who enjoy the full upside of any complementary good or service. Third, integration into the development function exposes the host to the costs and risks of developing complementary goods to the platform, especially in light of the inherent difficulty of anticipating end-user preferences in a consumer retail market.⁴⁹ Multiple factors suggest that those costs and risks can be especially high in the case of horizontal software applications (as distinguished from vertical applications customized for a particular use or industry⁵⁰) due to the complexity and volume of software programming, the scarcity of programming labor, the risk of product failure upon release, and the post-release costs of software "debugging", support and service. Fourth, integration by the host into any complementary goods market can discourage entry by third-party developers into that same market, thereby further inflating the host's integration costs and discouraging outside

⁴⁹ See Eisenmann et al., *supra* note 8.

⁵⁰ The lower costs of integrating forward into all aspects of software development in the case of a vertical product anticipates (correctly) that forfeiture practices, including open source development, is observed far less often in those sectors of the industry. Hence, it may not be coincidental that OSS has achieved success in the case of horizontal (or "platform") software technologies, such as operating systems, but more limited success in the case of vertical technologies targeted at a particular industry or user population. In the latter case, given the more constrained set of users and users, both contract specification and forward integration costs would be expected to be less onerous.

development that enhances platform value.⁵¹ Like contract, integration is therefore an important but imperfect (and often, extremely costly and even counterproductive) solution to the host's dilemma.

c. Forfeiture

There exists an elegant but draconian solution to the commitment problem: the host can forfeit ownership and/or control rights over the platform in whole or in part. Forfeiture practices can be understood broadly as encompassing any action that provides third parties with access to technological knowledge at any price below its market value, including zero or negative prices. While forfeiture enables the host to commit against opportunistic behavior, it violates the insolvency constraint to which even non-profit-motivated participants are subject. It is therefore necessary to reinstate control at some other point on the total consumption bundle, which in turn partially restores the host's dilemma.

(1) Forfeiture Strategies

Disclaiming ownership or control over the platform serves two purposes: (i) it gives users an asset that can not be expropriated by the host; and (ii) it allows users, or any other party, to enter into competition with the host. The host intentionally creates a potential entry threat, which then allows it to commit against opportunistic behavior *ex post* and elicit greater user adoption *ex ante*. Counterintuitively, the host secures market share by making its market share contestable.⁵² Potential rivals include users or any other

⁵¹ See Carliss Y. Baldwin & C. Jason Woodward, *The Architecture of Platforms: A Unified View*, in PLATFORMS, MARKETS AND INNOVATION 39 (Annabelle Gawer ed. 2009); Annabelle Gawer, *Platform Dynamics and Strategies: From Products to Services*, in PLATFORMS, MARKETS AND INNOVATION 57 (Annabelle Gawer ed., 2009); Gawer & Henderson, *supra* note 7.

⁵² Other commentators have identified rational incentives by monopolist sellers to invite competition. In the closest line of argument, Farrell and Gallini, argue that, when a consumer incurs sufficiently high "set up costs" a monopolist seller may (with a delay) rationally invite competition by "second source" producers in order to commit against higher second-period pricing that would expropriate the consumer's initial investment. See Joseph Farrell & Nancy Gallini, *Second-sourcing as a Commitment: Monopoly Incentives to Attract Competition*, 103 Q. J. ECON. 673 (1986). Shepard makes a similar argument with respect to second sourcing as a commitment to quality (in the form of delivery time) and Kende argues that a systems firm can increase profits by allowing competition in the aftermarket for secondary components, which constitutes a commitment to variety in the latter market. See Andrea Shepard, *Licensing to Enhance Demand for New Technologies*, 18 RAND J. ECON. 360 (1987); for a related argument, see Kende, *supra* note 47. In related arguments in a different context, Economides

party who can use the disclosed knowledge in order to develop competing technologies. For example, when a software provider releases a program's source code, it is exposed to "forking": that is, any user or group of users may develop non-compatible versions of the code that compete with and supplant the original version.⁵³ This describes the fate of AT&T's Unix after it was widely licensed: incompatible versions frustrated its broad acceptance in the market. But even more severe outcomes can be precipitated by forfeiture. As IBM discovered in its inability to maintain exclusivity over the PC, a firm that forfeits control over its core technology can lose the entire market to the producers of complementary goods within the consumption bundle. Less dramatically, Palm's decision to divest its operating system to a separate entity for widespread licensing—precisely in order to allay developers' concerns over host expropriation—did not yield favorable results.⁵⁴

Short of complete abandonment, the host can transfer control of the platform to a nonprofit organization, trade association or some other neutral entity over which the host lacks unilateral control.⁵⁵ This too was a strategy adopted by AT&T as well as other participants in the "Unix Wars" over setting the Unix standard in the 1980s: nonprofit

argues, and Katz and Shapiro suggest, that a monopolist will invite entry in order to commit to a high quantity that maximizes network effects and increases consumers' willingness to pay; otherwise, consumers will anticipate that the monopolist will constrain output in order to achieve supracompetitive pricing. *See* Economides, *supra* note 40; Michael Katz & Carl Shapiro, *Network Externalities, Competition and Compatibility*, 75 AM. ECON. REV. 242 (1985). Applying these lines of argument to the case of Intel, Gawer and Henderson argue that a monopolist platform holder has incentives to subsidize entry by third-party complementors in order to commit credibly against integrating forward into the complements market, which would then fail to maximize platform value. *See* Gawer & Henderson, *supra* note 7.

⁵³ This is actually the origin of the popular Firefox internet browser. As the program was being developed by the open source community using Mozilla code (which had been developed on a proprietary basis by Netscape but then released under an open source license following Netscape's commercial demise), two contributors who disagreed with the project's direction used the code to create the Firefox browser, which was ultimately adopted by community leaders as the primary supported version. *See* Jacques Bughin et al., *The Next Step in Open Innovation*, MCKINSEY QUARTERLY (2008), available at https://www.mckinseyquarterly.com/next_step_in_open_innovation_2155.

⁵⁴ *See* EVANS ET AL., *supra* note 22; Joseph Farrell & Phillip J. Weiser, *Modularity, Vertical Integration, and Open Access Policies: Towards a Convergence of Antitrust and Regulation in the Internet Age*, 17 HARV. J.L. & TECH. 85, 97-98, 100 (2003).

⁵⁵ *See* Stanley M. Besen & Joseph Farrell, *Choosing How to Compete: Strategies and Tactics in Standardization*, 8 J. ECON. PERSPECTIVES 117, 126 (1994) (on transfer to a neutral entity). In another variant, Gawer and Henderson observe that Intel attempts to commit against "squeezing" providers of complementary goods by establishing a not-for-profit unit (in corporate parlance, a "cost center") dedicated to generating and disseminating platform technologies among third-party complementors. *See* Gawer & Henderson, *supra* note 7.

organizations with publicly-interested-sounding names (the “Open Software Foundation”) enabled competing groups of hardware manufacturers to disclaim the ability to alter the accepted standard to the detriment of rivals and other users.⁵⁶ Since a nonprofit organization can not distribute net earnings to any outside controlling interest⁵⁷, it has reduced expropriation incentives and therefore an increased ability to elicit users’ investments. Through the vehicle of a nonprofit or other non-investor-owned organization, the host can commit against opportunistic behavior that (i) would only be rational in the case of an entity driven by profit maximization and (ii) can not be sufficiently excluded by contract. This conforms to a broader proposition advanced by Prof. Henry Hansmann: non-investor-owned forms of organization can be understood as a rational response to contracting failure resulting from informational asymmetries between transacting parties.⁵⁸

(2) *Forfeiture/Control Tradeoff*

Forfeiture is obviously the most potent means by which to commit against opportunism. As a stand-alone strategy, however, it is unworkable: the host can not capture revenues from forfeited portions of the platform. Broadly speaking, the host has two well-known options by which to render forfeiture an economically rational strategy consistent with the insolvency constraint to which even a non-profit-seeking entity is subject. First, in the case of partial forfeiture, it can secure revenues by regulating access to non-forfeited portions of the platform (or from user populations to whom access has not been forfeited). Second, even in the case of complete forfeiture, it can secure revenues from sales of complementary goods and services. The host therefore faces a basic tradeoff. On the one hand, it must forfeit control over a portion of the platform in order to elicit user adoption. On the other hand, it must exert control over some other portion of the platform, or some set of complementary goods or services, by which to

⁵⁶ For further discussion, see *infra* Part III.A.1.

⁵⁷ See BRUCE R. HOPKINS, *THE LAW OF TAX-EXEMPT ORGANIZATIONS* §§ 1.1(a), 20.1 (9th ed. 2007). Precisely, U.S. federal tax laws require that none of a nonprofit organization’s net earnings “inure” directly or indirectly to the benefit of any “individual or other person that has a close relationship with the organization” or “is in a position to exercise a significant degree of control over it.” See *id.* § 20.

⁵⁸ See Henry Hansmann, *The Role of Nonprofit Enterprise*, 89 *YALE L.J.* 835 (1980). These ideas are elaborated further in HENRY HANSMANN, *THE OWNERSHIP OF ENTERPRISE* (1996).

accrue revenues in order to cover development and maintenance costs (and, in the case of a for-profit entity, in order to cover costs and capture any remaining profits). That yields a preliminary hypothesis. Host entities will select hybrid organizational forms that partially or completely forfeit control over the platform in order to maximize adoption gains while retaining control in order to minimize revenue losses. I will now assess that hypothesis against observed organizational strategies in operating systems markets.

III. Organizational Convergence in Operating Systems Markets

Following transaction cost economics, the risk of opportunistic behavior given asset-specific investments can sometimes explain organizational and contracting structures observed in the market.⁵⁹ That hypothesis has been supported empirically in numerous circumstances.⁶⁰ Consistent with that approach, the risk of host opportunism, and the associated tradeoff between forfeiture and control, may be able to explain organizational forms in operating systems markets. The imperative to commit against host opportunism, subject to the insolvency constraint, provides a general explanation for the controlled use of forfeiture strategies, and the rough convergence of organizational forms used, by platform holders in these markets, irrespective of profit-seeking or non-profit-seeking motivations.

A. Old Models

To appreciate the economic motivations behind contemporary organizational forms in the operating systems market, it is useful to review briefly the leading “open” and “closed” structures that have been used historically in ICT and related software industries.⁶¹

⁵⁹ See Williamson, *supra* note 48.

⁶⁰ See CASE STUDIES IN CONTRACTING AND ORGANIZATION (Scott E. Masten ed., Oxford University Press 1996).

⁶¹ For sake of brevity, I mostly ignore the “mainframe” model used prior to the advent of the mass market for pre-packaged software and personal computing in the early 1980s. In that model, software was principally provided to business customers together with a hardware product (usually, a mainframe computer) without being separately priced and necessitated extensive customization and on-going support. See HILTZIK, *supra* note 6. I will briefly return to that model in my analysis of the normative implications of open models for platform developments, *see infra* Part IV. For a more detailed and comprehensive overview of various models of software development, *see* West, *supra* note 8.

1. The Unix Model: Software as Mostly Open Platform

The Unix operating system was developed in the early 1970s at Bell Labs. Its parent, AT&T, licensed Unix to universities at nominal cost and researchers developed a number of variants.⁶² In this model, the operating system acts as a “mostly open” platform, which spawns third-party development of applications that enhance its value. By the mid-1980s, Unix had become the industry platform in the minicomputer and workstation markets⁶³; however, its success was encumbered by the proliferation of incompatible versions. Starting in 1987, AT&T sought to standardize Unix in cooperation with Sun Microsystems, a workstation manufacturer, and then license it widely to chip manufacturers and system vendors (which it did so by 1989).⁶⁴ This was perceived as an attempt to “re-privatize” Unix, and, in 1988, IBM and other hardware manufacturers established the Open Software Foundation, a nonprofit entity that sought to standardize Unix and make it available on an open licensing basis.⁶⁵ AT&T subsequently sold its interests in the Unix operating system to Novell, a commercial firm, which transferred the rights to the UNIX trademark to the X/Open Group, Ltd., a nonprofit industry consortium dedicated to standardizing Unix systems.⁶⁶ These repeated forfeiture actions pursued a common objective: to place the operating system in the public domain, induce investments by developers and enable the recovery of rents through the closed hardware components of the total consumption bundle. As will be described subsequently, some of these hardware companies (notably IBM) are engaged in the same effort today—and are using the same foundation vehicle—to create an open operating system (based on Linux, a descendant of the Unix system) for the enterprise computing and smartphone markets.

⁶² See David McGowan, *Legal Implications of Open Source Software*, 2001 UNIV. ILL. L. REV. 241 [henceforth McGowan, *Legal Implications*].

⁶³ See MEYER & LEHNERD, *supra* note 31.

⁶⁴ See Garud & Kumaraswamy, *supra* note __.

⁶⁵ The formation of the Open Software Foundation may be more properly understood as an attempt to secure market share in the hardware market, as illustrated by the fact that some Open Software Foundation members introduced new Unix-based models shortly after its formation. *See id.*

⁶⁶ In 1996, X/Open merged with the Open Software Foundation to form the Open Group, a nonprofit association, which now holds the UNIX trademark and maintains the set of standards for operating systems that qualify as Unix. For further information, see “The Open Group—Unix”, avail. at <http://www.opengroup.org/press/23nov09.htm> (last visited Oct. 3, 2010).

2. The Windows Model: Software as Semi-Closed Platform

The modern personal computer industry is often dated from a transaction between IBM and Microsoft, wherein the latter agreed to provide the MS-DOS operating system (crucially, on a non-exclusive basis) for IBM's new personal computer, launched in 1981. The remaining sequence is well known. IBM was unable to preserve exclusivity over the PC, which became a commodity product cloned by other firms; IBM then attempted to preserve exclusivity over the new "premium" component, software, by developing the OS/2 operating system; it lost market share to Windows, the successor to MS-DOS, and exited the market in 1994. Microsoft retained exclusivity over the Windows platform, which persisted as a premium software product bundled with a premium microprocessor product sold by Intel and commodity hardware sold by many firms. But Microsoft voluntarily forfeited a portion of its legal monopoly: as discussed previously, it released APIs to independent software developers, who developed tens of thousands of applications (according to Microsoft, 80,000 applications as of 2008⁶⁷) that enhanced the value of Windows relative to its competitors. This giveaway was driven by an implicit contract: Microsoft forfeited technology and support services to developer-users, who generated complementary assets that supported platform value, thereby enabling Microsoft to earn revenues from end-users on the "other side" of the platform.⁶⁸ For Microsoft, prudent altruism has paid off handsomely: controlled forfeitures enabled it to overcome the host's dilemma and realize the network effects required to recoup its investment (and much more) in the platform and all extensions to it.

⁶⁷ See MICROSOFT, OPEN INNOVATION: THE EVOLUTION OF THE MICROSOFT APPROACH TO INTELLECTUAL PROPERTY (2008).

⁶⁸ It can even be argued that Microsoft engages in partial implicit forfeiture with respect to its sales of Windows to end-users, insofar as it charges a price well below the short-term profit-maximizing monopoly price. See David S. Evans & Richard Schmalensee, *Consumers Lose if Leading Firms are Smashed for Competing*, in DID MICROSOFT HARM CONSUMERS? TWO OPPOSING VIEWS (David S. Evans et al. eds., AEI-Brookings Joint Center for Regulatory Studies 2000); David S. Evans & Richard Schmalensee, *Be Nice to Your Rivals: How the Government is Selling an Antitrust Case Without Consumer Harm in United States v. Microsoft*, in DID MICROSOFT HARM CONSUMERS? TWO OPPOSING VIEWS (David S. Evans et al. eds., AEI-Brookings Joint Center for Regulatory Studies 2000); Richard Schmalensee, *Antitrust Issues in Schumpeterian Industries*, 90 AM. ECON. REV. 192 (2000). That amounts to a sacrifice of billions of dollars in profits annually and is consistent with the behavior of a repeat-player host entity.

B. New Models

From the introduction of Windows in 1985 through the widespread use of internet communications in the late-1990s, it would have appeared that the semi-closed model adopted by Microsoft had triumphed over the mostly open model that had promoted adoption of the Unix system. Microsoft set a uniform standard that governed virtually all of the Intel-based computing market while the Unix-based environment stalled amid an excessive number of variants. During roughly the past 10 years, however, (at least) two hybrid organizational models have emerged that obfuscate—or more precisely, *further* obfuscate—the distinction between open and closed systems in the operating system market. These models are: (i) the open source development of operating systems and other software applications by (in part) volunteer programmer communities, and (ii) open source development of operating systems and other software applications by commercial entities. Closer inspection shows that these models are often difficult to distinguish in practice: both tend to rely on some combination of corporate sponsorship to generate funding streams and nonprofit or related entities to manage and control core technological assets. Moreover, as will be shown, the role played by unpaid developers in OSS development is rapidly declining. This is consistent with theoretical expectations: the underlying tradeoff between platform forfeiture and control yields a common range of overlapping organizational structures that transcends ostensible differences in profit-seeking or non-profit-seeking motivations.

1. The Open Source Model: Software as Semi-Open Platform

The open source model departs most dramatically from proprietary software development through its uncompensated disclosure of source code subject to minimal contractual limitations. This apparent deviation from economic self-interest has attracted substantial attention from academic researchers. That scrutiny has in turn identified a mix of instrumentalist and non-instrumentalist motivations behind uncompensated developer contributions, where reputation effects and intrinsic interest appear to play a strong motivating role and ideological motivations play a weak motivating role.⁶⁹ The

⁶⁹ See Rishab Aiver Ghosh, *Understanding Free Software Developers: Findings from the FLOSS Study*, in *PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE* (Joseph Feller et al. eds. 2003); Eric von

apparent rational choice puzzle posed by OSS is overstated in light of a few key developments. Consider that (i) roughly half of open source programmers appear to be paid employees of corporate sponsors⁷⁰, who are repeatedly found to be the most productive contributors;⁷¹ (ii) some of the most successful open source applications follow a dual licensing model that relies on in-house code development and reserves technical support⁷² and some proprietary extensions for paying customers⁷³; and, as discussed below, (iii) the most successful open source applications depend on funding, personnel and other support supplied by proprietary sponsors. Standard characterizations of OSS development as the spontaneous coordination of ideologically motivated volunteers⁷⁴ do not accurately describe at least the most successful applications in the current market.⁷⁵ A publication in an IBM journal states this explicitly: “[T]he often

Hippel & Georg von Krogh, *Open Source Software and the ‘Private-Collective’ Innovation Model: Issues for Organization Science*, 14 *ORG. SCI.* 209 (2003); Karim R. Lakhani & Robert G. Wolf, *Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Project*, in *PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE* (Joseph Feller et al. eds. 2003); Josh Lerner & Jean Tirole, *Some Simple Economics of Open Source*, 50 *J. INDUSTR. ECON.* 192 (2002).

⁷⁰ See GREG KROAH-HARTMAN ET AL., *LINUX KERNEL DEVELOPMENT* (with respect to the Linux project); Ghosh, *supra* note 69.

⁷¹ See EVANGELIA BERDOU, *MANAGING THE BAZAAR: COMMERCIALIZATION AND PERIPHERAL PARTICIPATION IN MATURE, COMMUNITY-LED FREE OPEN SOURCE SOFTWARE PROJECTS* (dissertation submitted to London School of Economics and Political Science, 2007); Kroah-Hartman et al., *supra* note 70; Jeffrey A. Roberts et al., *Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects*, 52 *MGMT. SCI.* 984 (2006). In a case study of programmer contributions to the open source GNOME project, Berdou finds that paid programmers are more likely to contribute to and maintain critical portions of the code base; in the case of the open source KDE project, paid developers were more likely to maintain critical portions of the code base but no more likely to make contributions to those portions. See BERDOU, *supra* note 71 (on GNOME programmers); *id.* § 6.4.2 (on KDE project developers). Lancashire similarly finds that, based on per capita participation in the Linux and GNOME open source projects, open source projects tend to attract a disproportionate number of European relative to U.S. programmers, which correlates inversely with the relationship between programmer salaries in those regions. See Lancashire, *supra* note 26. The implication: consistent with a rational choice model, programmers who participate in open source projects are influenced by the opportunity cost of foregone alternative activities.

⁷² When an “open source” firm provides technical support to an enterprise customer, it will even prohibit modifications to the disclosed source code in order to maintain a stable code base on which support can reliably be provided. See Jim Markwith, *The Coexistence of Open Source and Proprietary Software*, PRACTICING LAW INSTITUTE (2008). That largely eliminates any meaningful difference from proprietary software.

⁷³ See 451 GROUP, *OPEN SOURCE IS NOT A BUSINESS MODEL: HOW VENDORS GENERATE REVENUE FROM OPEN SOURCE SOFTWARE* (Oct. 2008).

⁷⁴ See *supra* note 10.

⁷⁵ For other contributions casting doubt on the standard characterization of open source software, see 451 GROUP, *supra* note 73; Sebastian von Engelhardt & Stephen M. Maurer, *The New (Commercial) Open*

quoted notion that such [open source] software is written primarily by people working *gratis* for the general good is false . . .”⁷⁶ The substantial re-privatization of OSS development may disappoint its ideologically-inspired proponents; however, it is an unsurprising outcome given the forfeiture/control tradeoff that precludes any perfect resolution to the host’s dilemma.

a. Credible Commitment through Controlled Forfeiture

A host entity can use the following three devices to credibly commit against future opportunism: (i) giveaways subject to contract; (ii) norms/ideology; and (iii) a foundation entity or other nonprofit form of organization. The combination of these devices constitutes an “umbrella” contract that governs the relationship between the host and user populations, resulting in some intermediate level of control/forfeiture over the underlying platform and complementary set of goods and services. Contrary to conventional accounts, the imperative to commit against user expropriation, subject to the cost recovery constraint, can account for the forfeiture actions that characterize OSS development without any reference to altruistic motivations.

(1) *Contractual Giveaways*

The salient characteristic of OSS development is the uncompensated disclosure of source code subject to few contractual limitations. While often explained by reference to ideological motivations, this forfeiture action can be understood in instrumentalist terms

Source: Does It Really Improve Social Welfare? (Working Paper 2010). It is unclear whether this characterization ever had any basis in fact. A programmer survey released in 2000 found that open source code contributions rested on a narrow programmer base (10% of total authors wrote 72% of code) and that Sun Microsystems was one of the three leading institutional contributors of code. See Rishab Ghosh & Vipul Ved Prakash, *The Orbitten Free Software Survey*, 5 FIRST MONDAY (2000). Another survey released in 2003 found that, in the Apache open source project, a core group of approximately 15 developers were responsible for “almost all new functionality” added to the code, while a somewhat larger group were responsible for generating fixes to reported defects. See Audris Mockus et al., *Two Case Studies of Open Source Software Development: Apache and Mozilla*, 11 ACM TRANSACTIONS ON SOFTWARE ENG’G & METHODOLOGY 309 (2002). Interestingly, Mockus et al. find that the concentration of code contributions among developers in the Apache project was *greater* than the dispersion in selected commercial projects. See *id.* On reflection, that result may not be surprising: without wage incentives to ensure required effort, project management must rely on a smaller set of individuals that have accumulated sufficient reputational capital in order to be entrusted with making code contributions.

⁷⁶ See P.G. Capek, *A History of IBM’s Open-source Involvement and Strategy*, 44 IBM SYS. J. 249 n. 4 (2005).

as a mechanism by which to commit against host expropriation. This is especially vital in the case of an open source project, which, at least at its inception, has no resources by which to integrate forward into development and is therefore entirely reliant on developer-user contributions in order to establish platform value. By disclosing the source code, the host (which may be construed as being constituted by the founder, group of founders, or any other group of developers that can exercise some effective control over the project) limits its ability to expropriate developer-users' specific investments. This commitment is made irrevocable by the open source license, which enables users (or any rival entity) to freely copy, modify and distribute the released code and thereby exposes project management to market discipline for bad behavior.⁷⁷ Moreover, the General Public License ("GPL"), the most widely-used open source license, has a reciprocity clause that effectively protects any developer-user against expropriation by *other* developer-users. This clause obligates any user to distribute any derivative applications using the released code under the same "open source" terms as the original license⁷⁸, which ensures that (i) all developer-users have access to all derivative applications distributed by other developer-users and (ii) project management can not exploit user contributions in order to develop proprietary products to which access will be constrained.⁷⁹ These commitments are further bolstered by the costs of altering the terms of the license. Absent an agreement to the contrary, code contributors do not assign copyright to any collective entity, which means that changing the terms of the license

⁷⁷ For similar observations, see Egon Franck & Carola Jungwirth, *Reconciling Rent-Seekers and Donators – The Governance Structure of Open Source*, 7 J. MGMT. & GOVERNANCE 401 (2003); Merges, *supra* note 28; Siobhan O'Mahony, *The Emergence of a New Commercial Actor: Community Managed Software Projects* (2002) (dissertation submitted to Department of Management Science and Engineering Management, Stanford University) [henceforth O'Mahony, *Emergence of a New Commercial Actor*]; Joel West & Siobhan O'Mahony, *The Role of Participation Architecture in Growing Sponsored Open Source Communities*, INDUST. & INNOVATION (2008).

⁷⁸ See GNU, *GNU General Public License Version 2*, § 2(b) (1991), <http://www.gnu.org/licenses/gpl-2.0.txt>.

⁷⁹ Other open-source software is governed by more permissive licenses that place fewer or no constraints on the distribution of derivative applications. This has an ambiguous effect on inducing developer contributions. On the one hand, it may discourage those contributions by expanding the host's opportunities to expropriate user contributions for profit; on the other hand, it may encourage those contributions by expanding the opportunities available to developers by which to develop proprietary applications for profit. Note that, in the latter case, an outside developer's incentive structure is identical to that of a developer who develops applications for Windows.

under which all previous contributions were made would be prohibitively laborious.⁸⁰ Put differently: the transaction costs of contractual amendment enhance the commitment signal constituted by the license's substantive content.

(2) *Community Norms*

It is commonly stated that open source contributors are motivated by “community” norms that disclaim self-interested profit-seeking as distinguished from the profit-seeking behavior of market competitors. It is certainly the case that even mature open source projects such as the Linux kernel or the Firefox browser, which rely heavily on sponsored contributors (or, in the case of Firefox, paid employees) for core code development, continue to benefit from a mass of volunteers who contribute “bug reports” and a smaller group of volunteers who suggest “patches” to correct those defects.⁸¹ But a skeptical academic eye should consider whether the pronouncements of normative principle that accompany community contributions may be best understood as strategic tools by which to elicit a continuing flow of user contributions that are essential to project survival. That is: community norms may operate as an intermediate, rather than an ultimate determinant, variable in eliciting developer contributors.⁸² Without monetary or other material compensation by which to induce developer-user contributions, some mix of reputational and/or ideological rewards must take their place. The collectivist rhetoric that is characteristic of open source projects may therefore exert an economic function: the stigmatization of individual profit-seeking encourages contributions to a collective knowledge pool, which could otherwise be exploited for private gain. The moral tenor of principled rhetoric, as supplemented by reputational sanctions for shirking and reputational rewards for diligence⁸³, together mitigate the expropriation risk that would otherwise discourage user contributions. This is precisely why commercial firms that

⁸⁰ See O'Mahony, *Emergence of a New Commercial Actor*, *supra* note 77 at n. 49.

⁸¹ See Interview by Lenny T. Mendonca & Robert Sutton with Mitchell Baker, Chairman, Mozilla, in McKinsey Quarterly (Jan. 2008), *available at* http://www.mckinseyquarterly.com/Succeeding_at_open-source_innovation__An_interview_with_Mozillas_Mitchell_Baker_2098.

⁸² For a similar view, see Lancashire, *supra* note 26.

⁸³ See Lerner & Tirole, *supra* note 69, who provide evidence for the reputational mechanisms at work in the OSS setting.

employ open source strategies, and mature open source projects that rely heavily on corporate sponsorship, make strenuous efforts to develop and maintain a reputation for fairness and openness toward the developer community.⁸⁴ Consistent with the host's dilemma, any suggestion that an open source project will discriminate against community users—as occurred following Sun Microsystems' acquisition of the open source MySQL database in 2008 and recurred following Oracle's acquisition of Sun in 2010—prompts emotional protest and endangers the continuing flow of user contributions. This appears to be a subset of a general case that extends beyond the open source context: platform monopolists can overcome commitment difficulties by adopting rhetoric that reassures third-party providers of complementary applications that the host will not expropriate their investments in the platform.⁸⁵ In short: ideology reduces to strategy.

(3) *Foundation Entity*

Contrary to conventional accounts of a spontaneously-organized mass of volunteer contributors, it is clear that successful open source applications are governed by a strict hierarchy, in which a limited core of qualified developers develop code and approve changes to the code, who are in turn assisted by reports of defects and “fixes” contributed by a larger mass of participants.⁸⁶ But the organizational structure of open

⁸⁴ See Adam G. Cohn & Gary Spiegel, *Effective Open Source Development Practices*, in PRACTICING LAW INSTITUTE, PATENTS, COPYRIGHTS, TRADEMARKS AND LITERARY PROPERTY COURSE HANDBOOK SERIES 3, 5 (2009); Mario J. Madden, *Opening the Door: Four Questions to Ask in Developing an Open Source Software Policy* 7 (Practicing Law Institute 2006); Interview by Lenny T. Mendonca & Robert Sutton with Mitchell Baker, *supra* note 81.

⁸⁵ See Gawer & Henderson, *supra* note 7. To be clear, this is not to say that all pronouncements of publicly-interested values by open source participants or sponsors are cynically made for self-serving purposes; it is simply to say that these pronouncements will tend to proliferate as a solution to the commitment problem that afflicts the development of any platform technology that exposes users to expropriation risk. This is an observation made from the “anthropological” perspective of an outside observer, as contrasted with a the internal perspective of a direct participant. In the latter case, as Robert Frank has argued in a broader context, commitments to normative values are unlikely to avoid periodic temptations to defect unless they are internalized to some extent (which in turn would explain resistance to the outside observer's instrumentalist explanation). See ROBERT FRANK, *PASSIONS WITHIN REASON: THE STRATEGIC ROLE OF THE EMOTIONS* (1988).

⁸⁶ See STEVEN WEBER, *THE SUCCESS OF OPEN SOURCE* (2004); Andrea Bonaccorsi & Cristina Rossi, *Why Open Source Software Can Succeed*, 32 RESEARCH POLICY 1243 (2003) (on Apache and GNOME); IANSITI & LEVIEN, *supra* note 19 (on Linux); Lakhani & Wolf, *supra* note 69; McGowan, *Legal Implications*, *supra* note 62; Mockus et al., *supra* note 75 (on Apache and Mozilla); Roberts et al., *supra* note 71 (on Apache).

source projects has a key distinguishing element that gives them a competitive advantage over proprietary entities in committing against future opportunism. As shown in the Table below, nonprofit entities⁸⁷, which are subject to the control of advisory boards that are either self-appointing or elected by members⁸⁸, govern the development of core elements of the Linux-based operating system and suite of complementary open source applications. These include: (i) the Linux operating system⁸⁹ (which constitutes 21% of the worldwide market for operating systems for servers as of Q1 2010⁹⁰); (ii) the Ubuntu distribution (one of the leading noncommercial Linux distributions⁹¹); and (iii) the GNOME graphical user interface for use with Linux-based (and other Unix-based) operating systems. The same is true of other significant open source applications, including the Firefox browser, which constituted 24% of the worldwide browser market as of February 2010, and the Apache server application, which constituted 54% of the worldwide internet server market as of March 2010.⁹²

⁸⁷ For other discussions of foundation entities in the open source context, see O'Mahony, *Emergence of a New Commercial Actor*, *supra* note 77; Siobhan O'Mahony, *Guarding the Commons: How Community Managed Software Projects Protect Their Work*, 32 RESEARCH POLICY 1179 (2003) [henceforth O'Mahony, *Guarding the Commons*]; West & O'Mahony, *supra* note 77. While O'Mahony and West tend to view the nonprofit entity as an instrument by which the volunteer programmer community protects its interests against corporate encroachment, I view the nonprofit entity as an instrument by which corporate sponsors commit to programmers and other users that it will not act contrary to their interests.

⁸⁸ In a membership-based nonprofit entity, members have voting rights analogous to shareholders in a for-profit corporation but lack any rights to distributed earnings.

⁸⁹ Precisely, the Linux project concerns development of the "kernel", which refers to the central component of most operating systems that acts as a bridge between the application software and the hardware of a computer. There are multiple "distributions" of Linux-based operating systems. A distribution refers to a package consisting of an operating system, utilities and certain basic application programs.

⁹⁰ See Mary Jo Foley, *IDC: Windows Server still rules the server roost*, ZDNet.com, June 3, 2010 (describing results of market research released by International Data Corporation) (avail. at <http://www.zdnet.com/blog/microsoft/idc-windows-server-still-rules-the-server-roost/6424%7Ctitle=IDC>). Note that the figure above measures market share by number of server units sold. Note that Linux has a negligible market share in the desktop computing market, where Microsoft Windows continues to be the dominant system, but is the dominant operating system in the supercomputer market, which has always used Unix variants.

⁹¹ Note that there are other important Linux distributions; I have selected Ubuntu as a representative example of a leading noncommercial distribution (that is, it is not directly supported by a commercial distributor such as Red Hat or Novell). Even Ubuntu, however, is a boundary case as it is largely funded by Canonical, Ltd., which sells support services to the Ubuntu distribution.

⁹² See Netcraft, *April 2010 Web Server Survey* (April 2010), available at <http://news.netcraft.com/archives/category/web-server-survey/>.

Table II: Governance of Leading Open Source Applications⁹³

Product	Foundation (date est.)	License Type ⁹⁴	Governance Structure	Principal Sponsors (Board Members) ⁹⁵
Linux kernel	Linux Foundation (2007)	Reciprocal	Membership entity. Representation rights on board tied to membership dues. Maximum two board directors per member.	Fujitsu, Hitachi, Motorola, NEC, AMD, IBM, Intel, Oracle
Ubuntu (Linux distribution)	Ubuntu Foundation (2005)	Reciprocal, some caveats	Self-appointing Community Council and Technical Advisory board. Public commitment to zero royalty.	Canonical
Firefox browser	Mozilla Foundation (2003)	Permissive	Self-appointing board.	No formal corporate representation on board. As of 2008, Google contract provided in excess of 90% total revenues.
Apache server	Apache Foundation (1999)	Permissive	Membership entity. Governing board elected by members, who are admitted by majority vote of existing members.	No formal corporate representation on board. ⁹⁶
GNOME user interface	GNOME Foundation (2002)	Weakly reciprocal	Membership entity. Board of directors elected by members. No single entity may control more than 40% of board seats (including through affiliates).	Advisory board members: Canonical, Google, IBM, Intel, Mozilla Foundation, Nokia, Novell, Oracle, Red Hat.

⁹³ All foundation entities are 501(c)(3) tax-exempt organizations under U.S. federal tax law, except for the Ubuntu Foundation, which is organized as a not-for-profit entity under English law. Description of governance elements is based on the constituent documents of each foundation (as listed in the Appendix) and other information available on each entity's website.

⁹⁴ A reciprocal or "GPL" license refers to a license that contains a reciprocity clause that subjects all derivative products to the license's provisions; a permissive license refers to a license that lacks this reciprocity clause and is therefore amenable to the development of proprietary products based on the disclosed code; a "weakly reciprocal" license refers to a license that combines features of both license types and allows some latitude to combine disclosed code with proprietary files in derivative applications.

⁹⁵ For purposes of this column, a sponsor is a "principal sponsor" if it has a seat on the Foundation's board of directors, advisory board (in the case of the GNOME Foundation) or the technical advisory board. For information on Google's relationship with Mozilla, see MOZILLA FOUNDATION AND SUBSIDIARIES, INDEPENDENT AUDITORS' REPORT AND CONSOLIDATED FINANCIAL STATEMENTS (Dec. 31, 2008 and 2007), avail. at <http://www.mozilla.org/foundation/documents/mf-2008-audited-financial-statement.pdf>. All data on board membership based on information on each Foundation's website as of October 3, 2010.

⁹⁶ The Apache Foundation states that all board members are "individuals"; however, as of 2005, IBM reported that two of the nine members are IBM employees. See Capek et al., *supra* note 76. Based on research undertaken in July 2010, one member appears to be an IBM employee and one member appears to be a Google employee.

Other commentators have observed that a nonprofit entity is a useful logistical device for eliciting tax-deductible donations and providing a legal entity to hold intellectual property and other assets and enter into contracts and other legal relationships.⁹⁷ However its primary function may be to address the commitment problem that afflicts any host that seeks to elicit platform adoption. Placing core technology assets in a foundation entity binds project management to the constraints set forth in the foundation's charter, which in turn exposes the foundation to enforcement actions that could be undertaken by members, state regulatory authorities, and/or the Internal Revenue Service (which can revoke tax-exempt status⁹⁸). That in turn enables the host to induce the user investment required to sustain platform adoption. This is particularly urgent in the case of any open source application that relies on corporate sponsorship—as is the case in most leading open source applications today and, to some extent, all of the open source applications listed in the Table above. Developer-users fear expropriation given the substantial funding received directly or indirectly from corporate sponsors, each of which (as noted in the Table above) usually enjoys certain governance rights in the foundation.⁹⁹ The expropriation threat facing developer-users increases as the application achieves greater market success and the host incurs increasing opportunity costs by refraining from privatizing the application.¹⁰⁰ Re-privatization can be

⁹⁷ See O'Mahony, *Emergence of a New Community Actor*, *supra* note 77; O'Mahony, *Guarding the Commons*, *supra* note 87.

⁹⁸ This is not hypothetical. The IRS has initiated an inquiry into the tax-exempt status of the Mozilla Foundation given that it reportedly receives more than 90% of its revenues from "search royalty" payments pursuant to a contractual agreement with Google. The Google payments are made in consideration of Mozilla's agreement to make Google the default search engine in the Firefox browser. See Gregg Keizer, *Google Deal Produces 91% of Mozilla's Revenue*, PCWORLD (Nov. 19, 2008), available at http://www.pcworld.com/businesscenter/article/154198/google_deal_produces_91_of_mozillas_revenue.html.

⁹⁹ This commitment problem is nicely illustrated by a recent incident involving the openSUSE project, a leading Linux distribution. The developer community requested that the chief corporate sponsor, Novell, establish a foundation to provide a vendor-neutral governance mechanism to oversee future code development. This is despite the fact that the community is currently guided by an elected board of three Novell employees and two independent community contributors. See Ryan Paul, *openSUSE Linux Seeks Own Direction, More Autonomy from Novell*, ARS TECHNICA (2010), available at <http://arstechnica.com/open-source/news/2010/06/opensuse-project-seeks-feedback-on-strategy-drafts.ars>. If established, the openSUSE Foundation would constitute a not-for-profit foundation nested within the Linux ecosystem, which is itself governed by the nonprofit Linux Foundation.

¹⁰⁰ The acquisition prices paid for the most successful open source applications testifies to their commercial value: for example, Sun's acquisition in January 2008 of the MySQL open source database for \$1 billion; Yahoo's acquisition in September 2007 of the open source Zimbra collaboration software

accomplished through various means: restricting access to future code releases, limiting technical or other support, or transferring project control to an outside buyer who will have a rational profit interest in restricting access. If a single entity controls the rights to all code contributions (as would be the case if contributors were required to enter into an assignment agreement), then that entity could terminate or constrain the rights previously licensed under the GPL¹⁰¹ or, as is often the case in dual licensing models (the most popular distribution model among commercial open source entities¹⁰²), simultaneously license the code under a proprietary license that covers a software product with expanded functions and/or support.¹⁰³

Consistent with the role of the foundation as a commitment device, the Linux Foundation states that it is designed to support the independent development of the Linux system: “The Linux Foundation serves as a neutral spokesperson for Linux” and “It’s vitally important that Linux creator Linus Torvalds and other key kernel developers remain independent.”¹⁰⁴ Not accidentally, Linus Torvalds, the project’s founder,

product for \$350 million; and Novell’s acquisition in January 2004 of Linux SUSE, a leading Linux distributor, for \$210 million. *See Sun to Acquire MySQL*, MySQL (Jan. 16, 2008), available at <http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html> (on Sun’s acquisition of MySQL); *Yahoo! Announces Agreement to Acquire Zimbra*, ZIMBRA (Sep. 17, 2007), available at http://www.zimbra.com/about/zimbra_pr_2007-09-17.html (on Yahoo’s acquisition of Zimbra); *Novell Completes Acquisition of SUSE Linux*, NOVELL (Jan. 13, 2004), available at <http://www.novell.com/news/press/archive/2004/01/pr04003.html> (on Novell’s acquisition of Linux SUSE).

¹⁰¹ *See* McGowan, *Legal Implications*, *supra* note 62.

¹⁰² *See* 451 GROUP, *supra* note 73 (based on study of a selected set of 114 vendors); *see also* Cohn & Spiegel, *supra* note 84. In a dual licensing model, the application is distributed both in a free “community” version and a positively-priced “enterprise” version that has more features or more frequent updates and/or is accompanied by technical support.

¹⁰³ It might be unclear how a dual licensing strategy is consistent with the terms of an open source license. There are two scenarios. First, if a permissive license is used, then there is no obligation to license derivative applications under the same terms. Second, even if a reciprocal license is used (as in the popular GPL), the owner of the copyright to the code can elect to license the same code under both a reciprocal and proprietary licenses or license the “core” code base under a reciprocal license and provides an expanded code base under a proprietary license. That is supported by §2(b) of the GPL, which states that the license is not binding on the original author.

¹⁰⁴ This statement should be taken with a grain of salt. As noted elsewhere, *see supra* note __, Torvalds is an employee of the Foundation, which is in turn substantially governed by outside sponsors, which are therefore an indirect source of compensation. *See* LINUX FOUNDATION, ABOUT US (2010), available at <http://www.linuxfoundation.org/about>.

personally owns the Linux trademark¹⁰⁵, which therefore constrains the ability of any outside party to expropriate user contributions. This commitment against opportunism runs throughout the Foundation's bylaws, which both limit and disperse sponsors' governance rights. Some notable examples include: (i) contributing members receive certain rights to elect directors to the board but the charter limits the number of directors that are "monetarily compensated" by any member entity;¹⁰⁶ (ii) the Executive Director may not be an employee of any contributing member;¹⁰⁷ (iii) amendment to the bylaws or dissolution of the foundation requires the vote of a majority of the directors;¹⁰⁸ and (iv) no member can appoint more than two directors out of a required minimum of 10 directors.¹⁰⁹ Additionally, the Foundation's Advisory Board (which is controlled by sponsor-affiliated members) is not all-powerful; rather, certain powers are delegated to a Technical Advisory Board, End-User Council and Vendor Advisory Council, each of which operates subject to formal charters that specify membership requirements, committee powers and governance mechanisms that preserve some influence for selected (although somewhat overlapping) constituencies in the Linux community.¹¹⁰ This controlled diffusion of governance rights makes it difficult for any single sponsor, or for multiple sponsors to coordinate in order, to unilaterally direct foundation policy, change foundation governance or dissolve the foundation entity. The transaction costs of charter amendment are uncharacteristically welcome: they bolster the commitment signal constituted by the charter's substantive content.

¹⁰⁵ For the trademark registration, see <http://tess2.uspto.gov/bin/showfield?f=doc&state=4007:2huukf.3.217>.

¹⁰⁶ See LINUX FOUNDATION, AMENDED AND RESTATED BYLAWS OF THE LINUX FOUNDATION § 5.3 (2007), available at <http://www.linuxfoundation.org/about/bylaws>.

¹⁰⁷ See *id.* § 6.7 (2007).

¹⁰⁸ See *id.* § 9.2 (2007).

¹⁰⁹ See *id.* §§ 5.1, 5.3 (2007).

¹¹⁰ For further information, see LINUX FOUNDATION, COLLABORATE, avail. at <http://www.linuxfoundation.org/collaborate> (last visited April 21, 2010). It should be noted that the independence of the Technical Advisory Board and End-User Council is not entirely clear given that most members appear to be affiliated with corporate sponsors of Linux. Affiliation was determined based on stated affiliation, email address or information obtained through further internet searches, in each case as of June and July 2010.

b. Implicit Privatization: Is Linux a Subsidiary of IBM?

The umbrella contract between the host and users—constituted by contractual giveaways, community norms and the foundation charter—provides a powerful set of tools by which the host can commit against opportunistic behavior. But this solution is incomplete since it fails to provide any means by which to support platform development and maintenance costs (not to mention the above-cost return required in the case of a for-profit entity). There are four mechanisms by which an open source application can preserve a supporting revenue stream: (i) public subsidy funded by taxation, (ii) private subsidy in the form of philanthropy or other form of voluntary contribution, (iii) cross-subsidy through revenue streams from complementary goods, and (iv) price discrimination across user populations. The most successful open source applications appear to rely heavily on option (ii); in fact, it is really option (iii) in a not-so-subtle disguise.¹¹¹ In the following Table, I consolidate these funding mechanisms with the commitment mechanisms identified above, which then sets forth the complete set of organizational elements that may be combined to implement the forfeiture/control tradeoff in any institutional structure for developing and maintaining a platform good to which access is completely or substantially unconstrained.

Table III: Institutional Tools for Non-Exclusive Platform Design

<u>Commitment Mechanisms</u>	<u>Funding Mechanisms</u>
Giveaway by Contract	Private Subsidy (Gift)
Norms	Cross-Subsidy (Complementary Sales)
Foundation Entity	Price Discrimination

¹¹¹ For sake of brevity, I will omit option (i), which is largely inapposite as a practical matter, and option (iv), which has been discussed in the broader literature on pricing strategies in multi-sided markets. See, e.g., Nicholas Economides & Evangelos Katsamakas, *Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry*, 52 MGMT. SCI. 1057 (2006); Geoffrey G. Parker & Marshall W. Van Alstyne, *Two-Sided Network Effects: A Theory of Information Product Design*, 51 MGMT. SCI. 1494 (2005).

These elements can now be applied to analyze the organizational design of the Linux operating system. The result is starkly different from the standard characterization of open source projects in the legal (and even some of the economic) literature. Linux kernel development is overseen by the Linux Foundation, which is managed on a day-to-day basis by the project's founder, Linus Torvalds, who is employed by the Foundation. The Foundation is supported by cash contributions, personnel and other forms of support supplied by several corporate sponsors, each of which is entitled to representation on the Foundation's board based on the amount of its membership dues. The board consists of 12 representatives appointed by corporate sponsors and four independent representatives.¹¹² Sponsors' contributions are constituted by membership dues (total of \$4,455,000 in 2009)¹¹³ and sponsor personnel paid to contribute code and participate in the "sign-off process" on admitting new code into the Linux kernel. As shown in the Table below, paid corporate contributors account for a disproportionate share of both submitted changes (almost 80%) and sign-offs (over 85%). Note that the four leading sponsors are responsible for more than half of all sign-offs—a fact that is hardly consistent with a mass of spontaneously-organized volunteer contributors.

¹¹² Six large hardware firms are "Premium" members, each of which makes a \$500,000 annual contribution and is entitled to a seat on the Board while other firms at lower contribution levels collectively elect members to the Board. The collectively-elected board representatives are associated with Novell, Hewlett-Packard, AMD, NetApp and Motorola. All data on members found in LINUX FOUNDATION, MEMBERS, <http://www.linuxfoundation.org/about/members> (last visited April 21, 2010). Information on membership fees found in Linux Bylaws (see Appendix).

¹¹³ Figures reflect author's calculations based on data available on Linux Foundation website.

Table IV: Corporate Code Contributions to Linux Kernel Development (2009)¹¹⁴

Firm	% Code Changes	% Sign-Offs
Red Hat	12%	36.4%
IBM	6.3%	5.3%
Novell	6.1%	8.2%
Intel	6.0%	6.4%
Oracle	3.1%	1.2%
Fujitsu	1.5%	--
Google	0.8%	10.5%
Others	43.1%	17.4%
TOTAL	78.9%	85.3%

IBM in particular has made unilateral contributions to the Linux project (not to mention several other open source projects¹¹⁵) that go far beyond its membership dues and code contributions. In 2001, IBM pledged to provide \$1 billion in funding to Linux over the following three years (which it claims to have recouped by 2002¹¹⁶); in 2005, it (with other sponsors) founded (and donated 500 patents to) the Open Invention Network, an entity that purchases Linux-related patents and then licenses them on a royalty-free basis in order to protect developers from infringement claims; and, as of 2010, 10,000

¹¹⁴ All information based on data collected by the Linux Foundation. Figures shown reflect changes and sign-offs on Linux kernel version 2.6.25. See KROAH-HARTMAN ET AL., *supra* note 70. Note that these data assume conservatively that all contributions or sign-offs for which corporate affiliation could not be established were made by unpaid contributors.

¹¹⁵ Perhaps most notably, in 2001, IBM launched the Eclipse software development tool at a reported cost of \$40 million, which it paid to acquire code from a start-up; it then released it on an open source-based and, in 2005, it spun off the Eclipse group as an independent not-for-profit entity, the Eclipse Foundation. See LaMonica, *supra* note 12; McMillan, *supra* note 12; Taft, *supra* note 12. The release of Eclipse is tied to IBM's participation in Linux insofar as its platform-independent interoperability features enable Windows developers to write for the Linux platform. For further explanation, see David Berlind, *Open Source: IBM's Deadly Weapon*, ZDNET (2002), available at <http://www.zdnet.com/news/open-source-ibms-deadly-weapon/296366>. As indicated previously in Table III, IBM holds a board seat on the GNOME (user interface) foundation and an IBM employee appears to be a member of the board of the Apache Software Foundation.

¹¹⁶ It has not been possible to independently verify the \$1 billion investment made by IBM in Linux (that is, aside from announcements by IBM) or the returns that can be attributed exclusively or primarily to that investment, as IBM financial statements do not sufficiently break out the relevant data. However, it appears safe to say that IBM has made a substantial investment in Linux and has earned substantial returns on its investment; on the latter point, it clearly would have stopped making further investments if that were *not* the case. See Stephen Shankland, *IBM: Linux Investment Nearly Recouped*, CNET NEWS (Jan. 29, 2002), available at <http://news.cnet.com/2100-1001-825723.html>.

IBM employees were working in Linux-related positions in R&D, sales and marketing, including 600 developers at the Linux Technology Center.¹¹⁷ If we use the figure of \$69,620 as the national median salary for a programmer (as reported by the U.S. Bureau of Labor Statistics in 2009), then IBM's annual investment in the Linux Technology Center alone equals over \$41.7 million in salary expenses (not including overhead expenses). Most recently, in June 2010, IBM founded a nonprofit foundation, the Linaro Foundation, with major semiconductor manufacturers in order to develop software tools to advance third-party development of the Linux operating system for use on semiconductors used in smartphones, netbooks and other mobile computing devices.¹¹⁸

IBM's behavior may seem paradoxical: the world's leading patentee for the past 10 years is the leading contributor of cash, code and personnel to an enterprise that disclaims the use of patents and other forms of intellectual property. The economic rationale behind these lavish giveaways by IBM and other profit-seeking firms is easy to ascertain and rebuts the view that open source production provides an alternative to market production by profit-seeking entities. Viewed in the aggregate, the Linux project operates as a joint product development and marketing project that is fully funded, partially governed and partially operated by a commercial consortium that promotes adoption of an operating system platform by developer-users. Each member of this implicit consortium seeks to promote, and shares in the cost of promoting, a commodified platform in the form of an operating system that can promote sales of complementary hardware, other applications and/or warranty, support and consulting services to business end-users.¹¹⁹ These complementary goods and services fall into two categories. First, firms such as IBM sell Linux-compatible servers and other hardware. In a variant of this

¹¹⁷ See IBM SOFTWARE, LINUX AND OPEN SOURCE DRIVE INNOVATION (2010), available at <http://www-01.ibm.com/software/info/features/linux/main.html>.

¹¹⁸ See LINARO FOUNDATION, ARTICLES OF ASSOCIATION OF LINARO LIMITED (2010), available at <http://www.linaro.org/assets/PDF/LinaroArticlesofAssociation.pdf>. Other founding members are ARM Holdings, Freescale Semiconductor, Samsung, ST-Ericsson and Texas Instruments, all of which are semiconductor chip designers and/or manufacturers. For further information, see LINARO FOUNDATION, PRESS RELEASE, *ARM, Freescale, IBM, Samsung, ST-Ericsson and Texas Instruments Form New Company to Speed the Rollout of Linux-Based Devices*, avail. at <http://www.linaro.org/arm-freescale-ibm-samsung-st-ericsson-and-texas-instruments-form-new-company-to-speed-the-rollout-of-linux-based-devices/> (last visited June 15, 2010).

¹¹⁹ See Merges, *supra* note 28 (for similar observations); Joel West & Scott Gallagher, *Challenges of Open Innovation: The Paradox of Firm Investment in Open-Source Software*, 36 R&D MGMT. 319 (2006).

strategy, IBM and other firms sell software that embeds open source code in a software product that is sold on a proprietary basis (for example, the popular IBM WebSphere enterprise software suite, which builds a proprietary application on top of the Apache open source web server application¹²⁰). Second, firms such as Novell and Red Hat sell warranties, support services and subscriptions to Linux distributions, while maintaining unsupported community distributions that are available at no charge. For firms such as IBM, Red Hat and Novell¹²¹, the open source model provides a mechanism by which to commodify a platform technology, which induces outside development that enhances platform value, which in turn enables the sale of complementary goods and services.

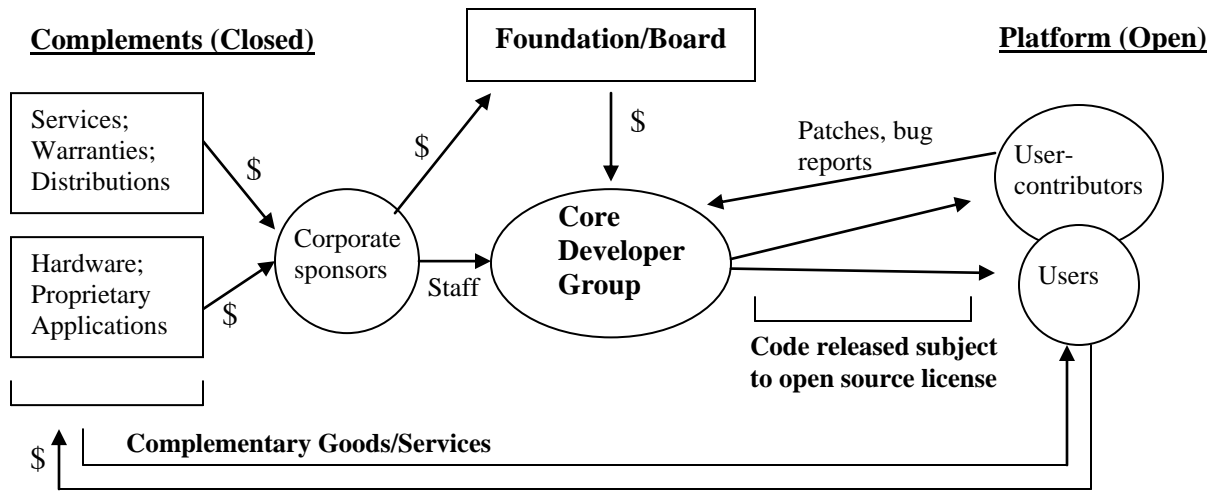
c. Summary: Forfeiture/Control Tradeoff (Again)

It is now possible to draw a full picture of the forfeiture/control tradeoff as implemented in institutional form in mature open source applications such as the Linux project. This is a mixed structure consisting of an open platform component and a closed complementary goods component. With respect to the open component, host opportunism is restricted by partially forfeiting control over the core technology, as implemented through contractual giveaways, norms and the foundation charter. With respect to the closed component, revenue streams are secured through subsidy mechanisms in the form of monetary and in-kind contributions by sponsoring entities, who support those investments by sales on complementary goods and/or services. As illustrated by cash flows in the Figure below, this arrangement is fully consistent with business rationality. So long as revenues from proprietary goods and services exceed contributions to platform maintenance and support, sponsoring entities can anticipate a net positive return through participation in the consortium.

¹²⁰ See IBM, IBM IS COMMITTED TO LINUX AND OPEN SOURCE (2010), available at http://ftp.software.ibm.com/linux/pdfs/IBM_and_Linux.pdf.

¹²¹ IBM explicitly describes its close relationship with the leading Linux distributors: “By working closely with Novell and Red Hat during all stages of development, IBM helps ensure that features needed . . . are included in the industry’s leading distributions.” See *id.* The link between IBM and Novell is especially close: in 2004, when Novell acquired SUSE Linux, a Linux distribution that most closely supports IBM processor-based servers and mainframes, IBM made a \$50 million equity investment in Novell in support of the acquisition. See Stephen Shankland, *Novell Inks Deal for IBM Servers, HP PCs*, CNET NEWS (Mar. 24, 2004), available at http://news.cnet.com/Novell-inks-deals-for-IBM-servers%2C-HP-PCs/2100-7344_3-5178475.html?tag=mncol;txt.

Figure II: Institutional Structure of Mature Open Source Systems



2. The New Proprietary Model: Competitive Forfeiture

Open models for software production start by addressing the credible commitment problem and must evolve to address the non-funding problem. Note that this is also true for non-profit-seeking enterprises that are subject to an insolvency constraint: absent any funding solution, the open model can not cover platform development and maintenance costs. Closed models for software production start with a solution to the non-funding problem and must evolve to address the credible commitment problem. Note that this is also true for profit-seeking enterprises that are subject to a more demanding profit-maximization constraint¹²²: absent a commitment device, the closed model can not elicit and/or maintain user adoption given host opportunism. Hence, economic self-interest can compel for-profit entities to forfeit knowledge assets even in the absence of any legal compulsion to do so. (Conversely, cost-feasibility constraints compel non-profit-seeking entities to impose access restrictions notwithstanding ideological aspirations to the contrary.) Below I show how for-profit competitors in the smartphone market—the site of the most recent battle to secure dominance for competing operating system

¹²² Following this constraint, the host will cease to forfeit access at the point where marginal gains attributable to user adoption no longer equal or exceed marginal costs in the form of lost revenues. By contrast, a non-profit-maximizing host will continue to forfeit access just up to the point where it is no longer able to cover platform development and maintenance costs.

platforms—are seeking to secure market share by forfeiting core technology assets to nonprofit or other non-investor-owned entities.

a. Nokia’s Gifts

Let’s return to, and expand upon, Nokia’s act of generosity with which this paper started. As shown in the Table below, Nokia has invested over \$700 million¹²³ in progressively acquiring full ownership of (and developing) the Symbian operating system. It then irrevocably transferred its interest to a nonprofit foundation governed jointly with its rivals in the handset manufacturing business and telecommunications and semiconductor firms that compete with Nokia in the mobile telecommunications market (see Table VI for current board members).¹²⁴ The foundation subsequently released the source code for the Symbian operating system to the public under an open-source license, while Nokia reportedly continues to make the bulk of code contributions.¹²⁵ But, as shown below, even this exceptional forfeiture action omits considerable sums invested by Nokia directly or indirectly to acquire and then give away valuable technologies relating to the Symbian project. In 2006, Symbian (and hence, Nokia indirectly) forfeited licensing revenues by selling its user interface technology to Sony Ericsson—a direct rival of Nokia in the handset market—in order to reduce any perception of undue control by Nokia (then the largest stakeholder in Symbian).¹²⁶ In 2008, Nokia invested \$153 million to acquire Trolltech, a firm that holds the rights to the open source “QT toolkit”, a

¹²³ This figure is calculated as follows: \$250 million to increase its ownership stake in Symbian Ltd. to 50% plus \$410 million acquisition cost to buy out remaining interests in Symbian Ltd. plus an estimated \$48 million cost to acquire the Psion operating system (as part of a joint acquisition involving Ericsson and Motorola). On the original acquisition transaction, see Matthew Rose et al., *Phone Giants Team Up to Challenge Microsoft*, WALL ST. J., June 25, 1998. This figure slightly understates Nokia’s investment as it does not include the cost it incurred in connection with the sale of Symbian’s user interface technology to Sony Ericsson (see *infra* note 128), for which the transaction price is not publicly available.

¹²⁴ Of 10 board members, Nokia only has one representative despite having invested 100% of the capital required to acquire the foundation’s only asset.

¹²⁵ See *Symbian to Fight Back on Market Share, Move to Open Source*, COMMUNICATIONS DAILY (July 8, 2009).

¹²⁶ See Kevin Fitchard, *Symbian Sheds UIQ*, TELEPHONY (2006). Sony Ericsson subsequently sold 50% of its stake to Motorola, which agreed, together with Sony Ericsson, to license the technology on equal terms to all mobile device vendors and invest in a developer support program. See Amy-Mae Elliott, *Sony Ericsson selling 50% of UIQ unit to Motorola*, POCKET-LINT, Oct. 15, 2007, avail. at <http://www.pocket-lint.com/news/10731/sony-ericsson-sells-uiq-motorola> (last visited October 3, 2010). This obviously tracks the forfeiture “solution” to the host’s dilemma.

popular cross-platform software development tool intended to facilitate third-party development of applications for the Symbian operating system. Toward this end, Nokia relicensed it under a more permissive license that enables third parties to use the toolkit to develop and distribute applications on a proprietary basis.¹²⁷ Unlike some of its competitors (for example, Apple), Nokia demands no portion of the developer’s revenues in return.

Table V: Nokia’s Gifts

Date	Action	Cost
1998	Nokia forms Symbian Ltd. with Motorola and Ericsson to acquire Psion operating system	c. \$48M
2004	Nokia increases ownership stake in Symbian to 50%	\$251M
2006	Symbian divests user interface technology to Sony Ericsson	Unknown
Jan. 2008	Nokia acquires Trolltech, owner of the Qt software development tool; relicenses it under more permissive license	\$153M
June 2008	Nokia acquires remaining interests in Symbian Ltd.; transfers interest to Symbian Foundation	\$411M
Feb. 2010	Symbian Foundation releases Symbian source code under open source license	n/a

Nokia’s record of generosity is consistent with the familiar pattern of host altruism. Nokia participates in a fierce competition for platform dominance where host entities must elicit developer investments without which scale can not be achieved and sustained. To secure market share for its operating system platform in the smartphone market, any host entity must commit to developers and other users (which, in this case, includes handset manufacturers and telecommunications operators with considerable bargaining power) that it has limited or no ability by which to expropriate user investments. Transferring core technologies to an independent nonprofit foundation, and

¹²⁷ See Ryan Paul, *Nokia Qt LGPL Switch Huge Win for Cross-Platform Development*, ARS TECHNICA (2009), available at <http://arstechnica.com/open-source/news/2009/01/nokia-qt-lgpl-switch-huge-win-for-cross-platform-development.ars>.

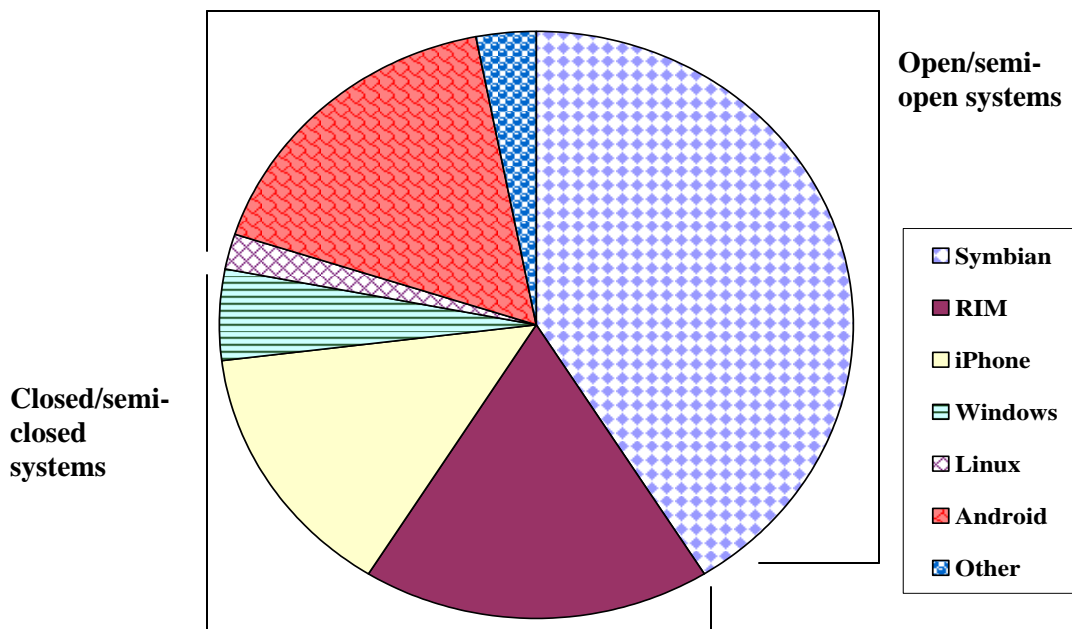
releasing those assets subject to an open-source license, are important steps toward this objective. It is worth noting the Symbian Foundation's cooperative architecture: its Board of Directors sits atop a federal structure consisting of a Feature and Roadmap Council, Architecture Council, User Interface Council and Release Council, each comprised of representatives from leading chip manufacturers, telecom operators and handset manufacturers.¹²⁸ Nokia is not alone in this organizational strategy. As shown below¹²⁹, over half of all operating systems used in smartphone devices worldwide are currently held or managed by nonprofit entities and/or have been released in part or in whole under an open source license without any royalty obligation.¹³⁰ Whether or not the market—and in particular, the independent developer population that is vital to any platform's success---will reward the generosity of Nokia and other like-minded host entities remains an open question.

¹²⁸ All information obtained from the Symbian Foundation website, which includes information on the representatives of each of the aforementioned bodies. For completeness, I note further that Nokia is also a partner with Intel in the development of MeeGo, an open-source Linux-based operating system platform designed for use in the higher-end segment of the smartphone market dominated by the closed systems of Apple (iPhone) and RIM (Blackberry). See Ryan Paul, *Nokia Picks MeeGo Linux, not Symbian, for Flagship Phones*, ARS TECHNICA (June 25, 2010).

¹²⁹ Figures are on a worldwide basis, as of Q2 2010. See GARTNER, PRESS RELEASE, *Gartner Says Worldwide Mobile Device Sales Grew 13.8 Percent in Second Quarter of 2010, But Competition Drove Prices Down* (Aug. 12, 2010). Market shares are more skewed when viewed on a more regional basis: proprietary systems have a larger market share in North America and Western Europe while non-proprietary systems (which are generally less expensive with fewer features) have larger market shares in East Asia and developing countries. Note that the lack of penetration into the North American market is most likely due to the fact that most Symbian-enabled phones are sold by Nokia, whose phones run on the GSM network standard, whereas most leading U.S. carriers operate on the CDMA network.

¹³⁰ These open systems include (i) the Symbian system, as described above; and (ii) the Linux-based Android system, promoted by Google, as described subsequently. Other Linux-based systems include Palm's webOS system, which includes open source and closed source components but is best described as a closed system, and the LiMo system, which includes open source and closed source components, and is best described as a semi-open system for reasons explained subsequently.

Figure III: Worldwide Market Share for Smartphone Operating Systems (Q2 2010)



b. Nonprofit Organization as Strategic Choice

The organizational structure of the market for operating systems for smartphone devices (clearly, a profit-seeking environment) closely mimics the organizational structure of the market for Linux-based operating systems for enterprise computing (ostensibly, a non-profit-seeking environment). While an implicit consortium supports the development of Linux for the enterprise computing market, explicit consortia support the development of open source operating systems for the smartphone market. Two consortia or similar arrangements coexist (and compete with) the Symbian Foundation. First, the 46-member LiMo (“Linux Mobile”) Foundation, a nonprofit trade association, was formed in 2007 by handset makers and telecommunications service providers in order to establish a non-proprietary Linux-based operating system for the smartphone market.¹³¹ The LiMo Foundation, whose members collectively represent over one billion

¹³¹ I say “non-proprietary” rather than “open source” because the LiMo Foundation restricts access to the source code behind its platform to firms who agree to the terms of an “IP safe harbor” which contains (among other things) a “non-assertion” obligation with respect to the intellectual property contained within the “common modules” of the platform. See LIMO FOUNDATION, AMENDED AND RESTATED BYLAWS OF THE LIMO FOUNDATION, Art. X.C (2009) (on IP safe harbor); *id.* Art. VI (on non-assertion obligation).

mobile telephone subscribers,¹³² operates subject to a detailed charter, which, in a manner similar to the structure of the Linux Foundation and the Symbian Foundation, disperses decisionmaking power over platform development and access policies across multiple committees, control over which is in turn allocated among groups of handset makers and telecommunications operators. Second, the 71-member Open Handset Alliance (“OHA”)¹³³, a loosely-organized association of handset makers, telecommunications service providers and other technology firms, led by Google, was formed in 2007 in order to promote the non-proprietary Linux-based Android operating system for the smartphone market.¹³⁴ Google reportedly controls development of the Android code (purchased by Google from a third party in 2005¹³⁵)¹³⁶, which appears to be developed in house by Google (in consultation with selected handset makers) and then released to the market under an open source license concurrently with the release of Android-compatible

Note, however, that (i) the LiMo platform is based on Linux and therefore in part encompasses open source code that is freely available; and (ii) the Foundation makes available to third-party developers application protocol interfaces on a royalty-free basis.

¹³² See *LiMo Foundation Seeks Alliance with WAC*, COMMWEB NEWS (Mar. 2, 2010).

¹³³ For a list of members, see OPEN HANDSET ALLIANCE, MEMBERS, http://www.openhandsetalliance.com/oha_members.html. Note that, unlike the other consortia, membership in the Open Handset Alliance is apparently free and does not appear to entitle members to any governance rights. As this appears to be a largely costless exercise, little importance should therefore be placed on the greater number of members in the Open Handset Alliance relative to the other consortia.

¹³⁴ I say “non-proprietary” rather than “open source” for a few reasons: (i) while the kernel code is released on an open source basis, Google only discloses it to outside developers concurrently with launch of the latest Android-supported handset; (ii) Google works on code development and other features with selected handset manufacturers prior to the official release date; and (iii) through agreements with carriers, Google incorporates into Android-enabled handsets a large number of proprietary Google applications. On Google’s work with handset manufacturers see John Biggs, *It’s Google’s World and Handset Makers Just Live in It*, CRUNCHGEAR (Nov. 14, 2009), available at <http://www.crunchgear.com/2009/11/14/its-googles-world-and-handset-makers-just-live-in-it/>. Of course, given the permissive nature of the open source license under which Android is released, carriers may elect to use the Android system without those applications.

¹³⁵ See GOOGLE, INC., ANNUAL REPORT (2009).

¹³⁶ See “Why is Google in charge of Android?” See ANDROID OPEN SOURCE, FREQUENTLY ASKED QUESTIONS, available at <http://source.android.com/faqs.html#aosp> (last visited June 16, 2010). The OHA’s governance structure is difficult to confirm due to the absence of formal constituting documents or publicly available contractual instruments; inquires to Google to obtain further information were not returned. However, OHA members reportedly have agreed to a “non-fragmentation agreement” whereby each member agrees not to support the development of separate, incompatible implementations of the Android source code. See *Google ‘Guarantees’ Android Compatibility*, ZDNET.CO.UK (Nov. 13, 2007), available at <http://news.zdnet.co.uk/communications/0,1000000085,39290713,00.htm>.

handsets.¹³⁷ Despite Google’s in-house code development, the Android platform does provide developers with the ability to download the source code together with a software development kit, which in turn facilitates development of complementary applications (as of July 2010, over 80,000 applications had been posted on the Android Marketplace).¹³⁸

If we consolidate membership in these three explicit consortia with the implicit consortium constituted by the 59-member Linux Foundation, as well as the explicit consortium constituted by the 6-member Linaro Foundation, a more complete picture emerges of the function played by nonprofit organizations in the competition for platform dominance in the smartphone device (and enterprise computing) markets. Remarkably, large swaths of vital intellectual property in the lucrative smartphone market are held by nonprofit consortia or related entities, which then sometimes release it with few contractual restrictions under an open source license. The Table below lists firms that hold board positions or are otherwise “material” participants in these consortia.¹³⁹

¹³⁷ OHA states on its website: “Google oversees the development of the core Android open source platform, and works to create robust developer and user communities”. The website also provides a lengthy answer to the question, “Why is Google in charge of Android?” See ANDROID OPEN SOURCE, FREQUENTLY ASKED QUESTIONS, available at <http://source.android.com/faqs.html#aosp> (last visited June 16, 2010). Note that the Android code appears to be developed almost entirely by Google programmers, as evidenced by a review of the list of leading contributors to the project (LIMO FOUNDATION 2009).

¹³⁸ See *Android Market Statistics*, ANDROLIB, <http://www.androlib.com/appstats.aspx> (last visited July 1, 2010) (on Android); *App Store Metrics*, 148APPS.BIZ, <http://148apps.biz/app-store-metrics/> (last visited July 1, 2010) (on Apple). That compares with 225,000 applications posted on the Apple App Store.

¹³⁹ With respect to each Foundation, I list a firm if it is a board member in any single foundation or a “material participant” in at least two foundations, based on information collected from each Foundation’s website in July 2010. If a firm is a board member in the case of any Foundation, that is indicated by “(B)”. A “material participant” is defined as follows: (i) for the Symbian Foundation, the firm is listed as a Founder or Core member (which is a function of membership dues), (ii) for the LiMo Foundation, the firm is listed as a Board member; and (iii) for the Linux Foundation, the firm is listed as a Platinum or Gold member (which is a function of membership dues). The Open Handset Alliance makes no membership distinctions and does not appear to require any membership fees; absent further information, it is difficult to assess the extent to which firms other than Google exercise a meaningful (if any) governance role. Despite not being a material member in any other consortium, I have added China Mobile, HTC, Sprint, T-Mobile, and Verizon as “material members” given that these entities (marked with an X*) appear to have played prominent roles in the initial manufacture and distribution of Android-enabled phones. For reasons of space, I omitted the following smaller firms: Aplix (a material member in the OHA and LiMo entities), NetApp (a board member in Linux), DeviceVM (a board member in Linux), and Wind River (a board member in LiMo and a member in OHA), which is now a subsidiary of Intel.

Table VI: Material Corporate Participants in Nonprofit Consortia Relating to Smartphone Market

	Firm/Foundation (Year est.)	Linux (2007)	Symbian (2007)	LiMo (2007)	OHA (2007)	Linaro (2010)
HANDBSET VENDORS	Fujitsu	X (B)	X (B)			
	Hitachi	X (B)				
	HTC				X*	
	LG			X (B)	X	
	Motorola	X (B)			X	
	NEC	X (B)		X (B)		
	Nokia	X	X (B)			
	Samsung		X (B)	X (B)	X	X (B)
	Sony Ericsson		X (B)		X	
TELECOMS	AT&T		X (B)			
	China Mobile				X*	
	NTT DoCoMo		X (B)	X (B)	X	
	Sprint				X*	
	Telefonica			X (B)	X	
	Verizon				X*	
	Vodafone		X (B)	X (B)	X	
	T-Mobile				X*	
CHIP VENDORS	AMD	X (B)				
	ARM					X(B)
	Freescale					X(B)
	IBM	X (B)				X (B)
	Intel	X (B)			X	
	Qualcomm		X (B)		X	
	ST Ericsson		X (B)		X	X (B)
	Texas Instruments		X (B)		X	X (B)
OTHER	Oracle	X (B)				
	Google	X			X	

It is worthwhile taking note of the types of firms that have elected to participate in these consortia: principally, handset makers, telecommunications providers and semiconductor chip providers. These are the holders of the components that, together with the operating system and software applications, constitute the consumption bundle delivered to end-users in the smartphone market. Using the Linux kernel, these holders of complementary assets have collectively integrated backwards into the operating system market. Each participant in these entities relinquishes its ability to secure revenues by regulating access to the platform, which is reduced to a utility privately operated in the form of an intermediate users' cooperative. This collective enterprise is consistent with its members' private interests. First, and most obviously, it avoids diverting rents to the stand-alone holders of "closed" operating systems (Microsoft, RIM, Apple).¹⁴⁰ Second, it results in cost savings relative to independent development of an operating system, which would impose both exorbitant direct costs in the form of development expenditures as well as indirect costs in the form of pricing discounts to reflect the increased risk of host opportunism. Third, and most importantly, the dispersion of control rights constrains each participant's ability to expropriate the platform-specific investments made by other participants as well as the broader population of developer-users who must be induced to invest in the platform. The absence of any controlling interest accelerates investments by these intermediate users in the operating system platform, which in turn enables participating firms to accrue revenues from intermediate users or end-users through the sales of other components in the consumption bundle. The LiMo Foundation states this objective explicitly: it seeks to pool technologies so as to create a common platform that enables its members to compete over the remaining differentiated portions of a smartphone device.¹⁴¹ Large-scale

¹⁴⁰ Even these systems are not entirely closed. Consistent with its historical practice, Microsoft releases Windows Mobile APIs to third-party developers while Apple enables the development of applications through its "App Store" (although, relative to Microsoft, it less generously demands that developers share with it 30% of revenues earned, sell the application exclusively through the App Store and reserves the right to reject submitted applications). This is actually a departure from Apple's original strategy to develop all applications internally consistent with *its* historical approach to developing complementary applications largely on an internal basis. See Kevin J. Boudreau & Karim R. Lakhani, *How to Manage Outside Innovation*, 50 MIT SLOAN MGMT. REV. 68 (2009). Similar initiatives have been undertaken (albeit less intensively) by RIM (Blackberry) and Palm.

¹⁴¹ See LiMO FOUNDATION, INTELLECTUAL PROPERTY POLICY (2009).

forfeiture of the most valuable technology assets in one of the world's most valuable markets is fully consistent with—indeed, it is *compelled* by—private self-interest in maximizing the rents derived from that market. But, if that is the case, then what is the *social* interest in facilitating those open practices? I now turn to that question.

IV. Implications: What's So Good About "Free"?

Two propositions are routinely asserted or implied in legal and economic (and some business) commentary on OSS and other open models of innovation: (i) open models are a novel departure from historically closed models of software and technological development; and (ii) open models are socially preferable and should be encouraged as a matter of public policy. The evidence set forth above shows that the descriptive proposition is false. As illustrated by the operating system market, ostensibly profit-motivated and non-profit-motivated participants often converge on a common range of organizational structures that mix open and closed elements in various permutations to achieve an efficient tradeoff between platform adoption and revenue accrual. Closed models often incorporate open elements in order to address opportunism concerns, while open models often—actually, absent public subsidy, *must*—incorporate closed elements in order to address solvency concerns. If there is often little meaningful (or, at least, no intrinsic) distinction between open and closed systems as a descriptive matter, then there may be reason to cast doubt on the normative presumption that the former should always be preferred as a policy matter over the latter. That presumption appears to drive policy proposals and, in some cases, government actions, to promote OSS adoption through subsidies, procurement mechanisms or other means.¹⁴² But, if “open” implies restrictions on access elsewhere in the total consumption bundle of products and services (in order to satisfy solvency concerns), while “closed” implies relaxations on access elsewhere in that same consumption bundle (in order to satisfy commitment concerns), then it is unclear why putatively open systems should be preferred over putatively closed systems. Both

¹⁴² For reviews of these policies, see GOVERNMENT POLICY TOWARDS OPEN SOURCE SOFTWARE (Robert W. Hahn ed., AEI-Brookings Joint Center for Regulatory Studies 2002); Engelhardt & Maurer, *supra* note 75, at §2.2; Jyh-Ann Lee, *New Perspectives on Public Goods Production: Policy Implications of Open Source Software*, 9 VAND. J. ENTMT & TECH. L. 45 (2007); Klaus M. Schmidt & Monika Schnitzer, *Policy Subsidies for Open Source? Some Economic Policy Issues of the Software Market*, CEPR DISCUSSION PAPER 3793 (2003).

the positive and normative assumptions that drive most legal, and a good deal of economic, discussion of open source models and other open forms of innovation are therefore subject to serious uncertainty. Open systems may yield no net social gain over closed systems and can impose a net social loss under certain circumstances.¹⁴³

The reason for the indifference result is by now familiar: market pressures will force open systems to close access at some point on the consumption bundle in order to satisfy solvency constraints; while market pressures will force closed systems to open access at some point on the consumption bundle in order to elicit user adoption. If putatively open and closed systems merely shift access restrictions from one point of the consumption bundle to another, then end-user welfare may be roughly equivalent under each environment¹⁴⁴, although the welfare of particular vendor populations may be substantially different. This can be easily illustrated. Assume a user is willing to pay a fixed price for a certain combination of software, hardware, operating system, network access and semiconductor chip: that is, the consumption bundle embodied by a smartphone handset. Of course it is true that software immediately costs less if it is available through free download under an open source license rather than being available at some positive price from a proprietary vendor.¹⁴⁵ But that simply enables the hardware

¹⁴³ Very recently, a nascent economic literature has begun to explore formal contingencies under which open source innovation models result in net social welfare gains and losses relative to proprietary models. The ensuing discussion above provides a verbal and more global analysis that derives from this Article's underlying tradeoff between commitment and solvency concerns; by contrast, the aforementioned contributions largely rely on a tradeoff between appropriability (roughly analogous to solvency constraints) and cost-spreading concerns, therefore omitting the commitment imperative. *See, e.g.*, von Engelhardt & Maurer, *supra* note 75; Gaston Llanes & Ramiro de Elejalde, *Industry Equilibrium with Open Source and Proprietary Firms*, Harvard Business School Working Paper No. 09-149, avail. at <http://www.ssrn.com/abstract=1425549>.

¹⁴⁴ Consistent with standard antitrust analysis, I am assuming that maximizing consumer welfare (or more precisely, in the technology context, end-user welfare) is the selected policy objective. Some scholarly commentators (but, rarely, courts) take the view that antitrust should be concerned with total welfare; however, either standard usually yields the same outcome, although there can be important exceptions (as in mergers that confer productive efficiencies in excess of allocative inefficiencies). Those discrepancies do not appear to be relevant to this discussion. For further discussion, see Eleanor Fox, *The Efficiency Paradox*, in *HOW THE CHICAGO SCHOOL OVERSHOT THE MARKET: THE EFFECT OF CONSERVATIVE ECONOMIC ANALYSIS ON ANTITRUST* 78-79 (Robert Pitofsky ed., 2008).

¹⁴⁵ The qualifier, "immediately", is necessary because even open source software is never free past the initial acquisition step. Software available for free download under an open source license lacks support and other functions that increase usability, which then compels the user to incur those support costs either internally or through the services of an outside provider. It is therefore an open question whether OSS is always the least-cost option taking into account the total cost of ownership, especially given that maintenance costs are the far greater portion of total software costs. *See* LIMO FOUNDATION 2009, *supra*

vendor (for example) to charge more—in fact, to charge exactly the amount the user has “saved” through the free download of software. No party has gained or lost in that scenario except the software vendor (lost) and the hardware vendor (gained): user welfare remains constant. This simple case illustrates a broader point. Apparently publicly-interested arguments in favor of open platforms—the rhetoric favored by the explicit and implicit industry consortia that promote OSS adoption in the enterprise computing and smartphone markets—may sometimes reduce (at least in part) to privately-interested arguments in favor of shifting rents from one particular set of industry players to another. Given that there would appear to be no distributional argument that rents should accrue to hardware providers (IBM) over software providers (Microsoft) (as in the enterprise operating systems market), or to handset makers (Nokia) and telecommunications providers (Vodafone) over integrated operating system and handset vendors (Apple) (as in the smartphone operating systems market), the use of “open” or “closed” models for intellectual production is a competitive choice bereft of policy implications that would invite state intervention to direct market outcomes.¹⁴⁶

It is even possible to suppose reasonable circumstances where public interest would favor a closed model over an open model with respect to any individual component of any given consumption bundle. Let’s suppose a market where firm *A* distributes software for free, either by choice or due to the lack of any legal or

note __; David S. Evans, *Politics and Programming: Governmental Preferences for Promoting Open Source Software*, in GOVERNMENT POLICY TOWARDS OPEN SOURCE SOFTWARE (Robert W. Hahn ed., AEI-Brookings Joint Center for Regulatory Studies 2002). For evidence that the total cost of ownership of open source applications sometimes appears to exceed the proprietary alternative, see Schmidt & Schnitzer, *supra* note 142, at 496-97.

¹⁴⁶ Precisely, I am referring to *publicly*-interested state interventions that seek to promote consumer welfare, consistent with the conventional antitrust standard referenced above. As a practical matter, it might be the case that a real-world government would elect to intervene in order to promote a rent transfer to firms that either (i) have disproportionate influence over the political process, or (ii) have a competitive advantage in the government’s jurisdiction relative to firms in other jurisdictions. The latter consideration may explain why European governments appear to be among the most vigorous proponents of public measures to advance open-source software adoption. This may simply be a form of technological protectionism: given the absence of any major European software provider with the exception of Germany’s SAP (which in turn may reflect the limited availability of patent protection for software in Europe, higher software piracy rates, and wider legal safe harbors for reverse-engineering software), European governments may seek to promote competition at other points on the ICT consumption bundle where domestic firms may be able to enjoy greater commercial success. On European policies favoring OSS, see Lee, *supra* note 143. Alternatively, it could be argued that the relative paucity of U.S. government initiatives favoring OSS procurement reflect the political influence of the subset of proprietary firms that currently enjoy market success based on closed business models.

technological means by which to regulate access. It must then secure a funding stream in order to cover its development and maintenance costs; let's assume it can do so through sales of complementary hardware products. That roughly describes the position of IBM in the mainframe market that preceded the advent of the market for personal computers and pre-packaged software: it bundled software with a hardware computing device and did not price the software separately.¹⁴⁷ Now suppose further that firm *A* has a dominant position in the hardware market (e.g., IBM in the mainframe era) and firm *B* (e.g., a start-up called Microsoft) has—or more precisely, could develop—a software product but has no immediate ability to produce a complementary hardware product.¹⁴⁸ As the provider of a complementary software product, firm *B* has a positive but reduced incentive to invest in development because it faces two unattractive entry opportunities: (i) it can contract with firm *A*, who will appropriate a substantial portion of revenues on the total consumption bundle, reflecting its dominance of the hardware market; or (ii) it can incur the (potentially high to exorbitant) fixed costs of developing the capacity to produce or otherwise deliver a complementary product over which it can regulate access and earn positive revenues. That explains why no firm can easily enter the market today *only* with a browser technology (Google, Mozilla and Microsoft supply it for free) or a search engine technology (Google, Microsoft and others supply it for free).¹⁴⁹ The result: using an open model to distribute any given component in the total consumption bundle—in our stylized example, software—reduces competition in the market for that component by compelling any entrant to incur the costs of simultaneously delivering some other complementary good in order to earn positive revenues.¹⁵⁰ That *absolute* increase in entry costs constitutes an entry barrier to the extent that incumbents, as compared to

¹⁴⁷ See FRANKLIN M. FISHER ET AL., FOLDED, SPINDLED AND MUTILATED: ECONOMIC ANALYSIS AND U.S. v. IBM (1983).

¹⁴⁸ The scenario need not be so extreme to arrive at the ultimate result: it must simply be the case that firm *A* produces complementary hardware at a lower cost than firm *B*.

¹⁴⁹ More precisely, it should be said that no third party can enter those markets without a significantly superior product for which consumers in those markets would be willing to pay a positive price.

¹⁵⁰ Ronald Mann has similarly observed that the rise of OSS can support industry consolidation by privileging the holders of complementary assets that must be used to capture returns on unprotected assets. See Ronald Mann, *Commercializing Open Source Software: Do Property Rights Still Matter?*, 20 HARV. J. L. & TECH. 1, 32-33 (2006).

entrants reliant on external financing, bear lower *relative* costs of capital in funding those costs.¹⁵¹

This argument is substantially consistent with observed trends in the evolution of OSS development. As abundantly illustrated by the above-described data on code contributions, funding sources, and board membership, a relatively concentrated group of hardware makers, semiconductor firms, and telecommunications operators dominate funding, governance and staffing of leading OSS projects in the enterprise and mobile computing markets. Code giveaways promote the economic fortunes of firms that have a competitive advantage in some other set of complementary assets—and conversely, operate to the economic detriment of firms that do not. Hence, IBM’s sponsorship of the Linux project and other open source projects that together form a larger open-source ecosystem independent of the “Wintel” (Windows operating system *plus* Intel chips) platform reinstates in part its old mainframe model.¹⁵² IBM *again* gives away software, for which it has incurred more than a billion dollars in aggregate development costs, in order to sell proprietary hardware (plus inherently excludable support services and proprietary software), for which it presumably earns a revenue stream in an equal or greater amount.

That may or may not be good for end-users, who may be indifferent among states of the world that re-distribute the aggregate pool of industry rents among the holders of dominant positions in different segments of the consumption bundle, or, at least in the case of developer-users, may reap short-term benefits from the virtuous race among competing platform holders to release control over technological assets in order to induce developer adoption. However, it is almost certainly *not* good for at least some stand-alone entrepreneurs in the operating system (or whatever may be the “free”) component

¹⁵¹ For discussion of why this is a reasonable assumption, see Williamson, *supra* note 48. Note that the entry-deterrent effect derives from the differential capital costs of incumbents and entrants, rather than the absolute increase in entry costs attributable to the commoditization effect in an open innovation market. That assumption renders this proposition compatible with the narrowest “Chicago school” definition of entry barrier as a cost that must be incurred by an entrant but not by an incumbent.

¹⁵² I say “reinstate *in part*” because there is a key difference: the OSS subsidized by IBM today can work across a variety of hardware devices sold by different vendors; by contrast, the software distributed by IBM in the mainframe era could only run on its hardware. That difference both reduces IBM’s total required investment placed at stake but exposes it to greater potential competition in complementary goods markets (which in turn enhances its ability to commit against expropriating user investments).

of the consumption bundle, who face higher entry costs given the *de facto* requirement to supply at least one other component in order to recover costs on the “free” component.¹⁵³ That inherent exclusionary effect casts a different light on the nonprofit consortia that provide competing operating systems for the smartphone market. Commodifying the operating system enables these consortia to “squeeze” the bargaining power of proprietary holders of stand-alone operating systems, who will then be compelled to forfeit some portion of (or even all) industry rents to the holders of the remaining assets that make up the rest of the consumption bundle.¹⁵⁴ If that strategy is successful, it is not clear whether smartphone end-users would be worse or better off. They may be better off if commodifying the platform technology promotes entry in the market for complementary goods and services by relieving those entrants from paying license fees to an operating system holder. That is what happened in the hardware market following IBM’s unintentional commodification of the personal computer—a result that promoted the mass distribution of personal computing devices. But the latter possibility can not be excluded. Depending on competitive conditions in the remaining portions of the supply chain, commodification of the operating system might expose end-users to the enhanced pricing power of firms that have competitive advantages in other segments of the consumption bundle. That is the other half of the history of the personal computing market: as a result of IBM’s unintentional commodification of the hardware component,

¹⁵³ I say “at least some entrepreneurs” because start-ups can (and do) exploit commercial open source models; however, this usually relies on a form of dual licensing, which, where successful, tends to result in acquisition or investment by a larger proprietary firm. *See* 451 GROUP, *supra* note 73. Moreover, it is still the case that the inability to regulate access over the primary knowledge good (in this case, software) may impede competition by foreclosing proprietary vertical strategies that focus on supplying that good to meet a specific market segment. Moreover, even in a world where access *can* be regulated, there does not appear to be any inherent impediment to undertaking an open source strategy if that were the profit-maximizing option. As illustrated by the widespread giveaways in the ICT industry, it appears that firms would often elect to relinquish access voluntarily. There is, however, one (potential) caveat to this optimism: incumbents that rely on patent protection to implement a closed business model may use those property rights to impede entry by open source rivals. A recent flurry of infringement claims against Google’s use of Android operating system in smartphone handsets appears to illustrate this tactic. *See* Ashlee Vance, *Microsoft Activates Android Lawsuit Against Motorola*, N.Y. TIMES, Oct. 1, 2010. To be clear, however, it is important to resist the natural inclination that any patent-infringement lawsuit against an open innovation enterprise is inherently anti-competitive or detrimental to user welfare; the contingencies explored immediately above suggest that the contrary interpretation can not be excluded and, under certain circumstances, is the most plausible outcome.

¹⁵⁴ Note that the same characterization may apply to competing efforts by hardware manufacturers in the 1980s to standardize the Unix operating system on an “open” basis, as described previously. *See supra* notes ___ and accompanying text.

Microsoft and Intel established dominant positions in the operating system and microprocessor portions of the “IT stack”, respectively. It is unclear whether shifting rents from IBM, as a dominant provider of proprietary hardware, to “Wintel”, as the dominant provider of operating system and chip technology, improved, degraded or had no effect on end-user welfare.

That would seem to leave us in an analytically unsatisfying state of indeterminacy. Let’s see if a different outcome results if we suppose that neither firm A nor any other firm has a dominant position in the complementary hardware market. If that is the case, then the free distribution of software would appear to result in a net improvement in consumer welfare (at least as a static matter): consumers would enjoy greater access to software without enhancing the market power of vendors in any complementary goods market. Now commodification of the software component in the consumption bundle is socially immaterial because it has not resulted in any countervailing increase in pricing power in any complementary goods market. Unfortunately that state of affairs is either unstable or infeasible. If neither the hardware market nor the software market—let’s reduce the consumption bundle to two components for simplicity—would allow any firm to capture anything more than a competitive return, no firm could recover its fixed R&D costs and would therefore decline to make further innovations. To garner a pricing advantage sufficient to support those fixed costs, any firm will seek to establish a protected position with respect to some component in the total consumption bundle. That is: if the hardware market is competitive, then firms will not select a free distribution strategy in the software market or, if law or technology makes it impossible to regulate access to the software component, then firms will decline to enter altogether. That effectively restores the indifference result: if firms can not achieve pricing power in any portion of the consumption bundle, then either (i) commodifying the software component would not result in any long-lasting marginal improvement in consumer welfare (since the funding stream would disappear and the component would be re-privatized) or, by anticipation, (ii) no firm would make any effort to commodify the software market in the first place.

But there is one contingency where an open model may improve the welfare of certain user populations even in the long term, which is consistent with the standard

presumption in favor of open over closed models—albeit over a substantially reduced scope of application. Let’s suppose that a firm has a sufficiently dominant position in the enterprise hardware market such that it is willing to release complementary software at a reduced or zero price in order to enhance sales of the hardware product. Now suppose there are two classes of users: (i) large-enterprise user *A*, who must obtain both the software and the enterprise hardware in order to achieve its desired objective, and (ii) sophisticated individual and/or small-enterprise user *B*, who must obtain the software but does not require the enterprise hardware to achieve its desired objective (for example, suppose the user can run the free software on commodified personal computing hardware). For user *A*, the indifference thesis proposed above holds: it pays roughly the same aggregate price across markets but simply allocates it differently among firms. However, for user *B*, free software results in an aggregate reduction in its total price burden because it never pays the enterprise hardware provider: the large enterprise user (who must purchase the complementary hardware product) is effectively subsidizing use by the sophisticated individual or small business user. In this case, free software might still not yield any aggregate efficiency gain but it does yield a user-to-user distributional gain that may be socially relevant: namely, it enables the provider to charge a higher price to enterprise user *A*, which in turn enables free distribution to sophisticated individual and/or small-enterprise user *B*. Whether or not that distributional transfer matters from a social perspective does not appear to give rise to any firm conclusions that would clearly call for state intervention. Even if that distributional transfer *does* matter from a social perspective, establishing a policy case for affirmatively promoting open innovation models still requires identifying impediments that would prevent the market from reaching that outcome independently.

Conclusion

Recent commentary among legal and some economic scholars has focused on apparently anomalous deviations from economic rationality in open source software markets where individuals or firms apparently give away substantial time, labor and technology. These accounts understate the incidence and rationality of giveaway practices. In particular, these accounts, which rely heavily on non-economic factors as explanatory variables,

overlook the use of access giveaways by commercial entities and the use of access controls by (ostensibly) non-commercial entities. The economic problem described by the host's dilemma can account for the mixed use of access giveaways and access controls irrespective of commercial or noncommercial motivation. The inherent tradeoff between inducing user adoption and preserving cost recovery anticipates intermediate structures that mix open and closed access policies over the total consumption bundle constituted by the platform and complementary goods and services. That tradeoff explains why the most dominant firms have regularly given away some of their most valuable technologies and why those firms now sponsor the development of operating systems that are available to users and rivals. The market rewards generosity: to win a platform race, the clever host must leave a substantial portion of total revenues to third parties that provide complementary goods. Conversely, the market punishes the selfish host who keeps too large a portion of market revenues for itself. But the market rewards prudence too: without exerting some control at some point of the consumption bundle, the host recklessly violates the insolvency constraint and platform demise ensues. Curiously the most valuable technologies may sometimes be the most difficult to commercialize.

APPENDIX: FOUNDATION DOCUMENTS

Apache Software Foundation

Certificate of Incorporation. March 25, 1999. Avail. at <http://www.apache.org/foundation/records/certificate.html>.

Bylaws of the Apache Software Foundation. June 1, 1999. Avail. at <http://www.apache.org/foundation/bylaws.html>.

GNOME Foundation

Bylaws of Gnome Foundation. April 5, 2002. Avail. at <http://foundation.gnome.org/about/bylaws.pdf>.

LiMo Foundation

Amended and Restated Bylaws of the LiMo Foundation. September 28, 2009. Avail. at http://www.limofoundation.org/images/stories/pdf/090928_pub_limo_bylaws_consolidated_as_of_september_28th_09.pdf.

Linaro Foundation

Articles of Association of Linaro Limited. May 2010. Avail. at <http://www.linaro.org/assets/PDF/LinaroArticlesofAssociation.pdf>.

Membership Rules of Linaro Limited. May 2010. Avail. at <http://www.linaro.org/assets/PDF/MembershipRulesofLinaroLimited.pdf>.

Linux Foundation

Amended and Restated Bylaws of the Linux Foundation. August 9, 2007. Avail. at <http://www.linuxfoundation.org/about/bylaws>.

Mozilla Foundation; Mozilla Corporation

Articles of Incorporation. July 14, 2003. Avail. at <http://www.mozilla.org/foundation/documents/mf-articles-of-incorporation.pdf>.

Bylaws of Mozilla Foundation. Avail. at <http://www.mozilla.org/foundation/documents/mf-bylaws.pdf>.

Bylaws of M.F. Technologies. Avail. at <http://www.mozilla.com/en-US/legal/bylaws.html>.

Symbian Foundation

Articles of Association of Symbian Foundation Limited. February 26, 2009. Avail. at <http://www.symbian.org/sites/default/files/07%20-%20SF%20Memorandum%20and%20Articles.pdf>.

Memorandum of Association of Symbian Foundation Limited. February 26, 2009. Avail. at <http://www.symbian.org/sites/default/files/07%20-%20SF%20Memorandum%20and%20Articles.pdf>.

Symbian Foundation Membership Rules. 2009. Avail. at <http://www.symbian.org/sites/default/files/06%20-%20SF%20Membership%20Rules%20v1.1.pdf> (last visited April 26, 2010).