

ARTICLES

THE HOST'S DILEMMA: STRATEGIC FORFEITURE IN PLATFORM MARKETS FOR INFORMATIONAL GOODS

Jonathan M. Barnett

CONTENTS

I. VOLUNTARY FORFEITURE: A TYPICAL PRACTICE	1869
A. <i>Bell Labs: Open Licensing</i>	1870
B. <i>Microsoft: Application Programming Interfaces</i>	1872
II. VOLUNTARY FORFEITURE: A RATIONAL PRACTICE	1874
A. <i>Some Economics of Platform Markets</i>	1875
B. <i>The Double Commitment Problem</i>	1878
1. <i>The Intertemporal Dilemma</i>	1878
2. <i>The Host's Dilemma</i>	1879
(a) <i>The Simple Case</i>	1879
(b) <i>The Complex Case</i>	1881
C. <i>Solutions to the Host's Dilemma</i>	1884
1. <i>Contract</i>	1884
2. <i>Integration</i>	1885
3. <i>Forfeiture</i>	1887
(a) <i>Simple Forfeiture</i>	1887
(b) <i>Complex Forfeiture</i>	1888
III. ORGANIZATIONAL CONVERGENCE IN OPERATING SYSTEMS MARKETS	1890
A. <i>Old Models</i>	1891
1. <i>The Unix Model: Software as a Mostly Open Platform</i>	1891
2. <i>The Windows Model: Software as Semi-Closed Platform</i>	1892
B. <i>New Models</i>	1893
1. <i>The Open Source Model: Software as Semi-Open Platform</i>	1893
(a) <i>Credible Commitment Through Controlled Forfeiture</i>	1896
(i) <i>Contractual Giveaways</i>	1896
(ii) <i>Community Norms</i>	1898
(iii) <i>Foundation Entity</i>	1899
(b) <i>Funding Controlled Forfeiture (or, Is Linux a Subsidiary of IBM?)</i>	1906
2. <i>The Nokia/Google Model: Software as Semi-Open Platform</i>	1913
(a) <i>Nokia's Gifts (and Regrets)</i>	1914
(b) <i>Nonprofit Organization as Strategic Choice</i>	1920

IV. IMPLICATIONS: WHAT'S SO GOOD ABOUT "FREE"?	1925
A. <i>The Indifference Baseline</i>	1927
B. <i>The Case for Closed Models</i>	1929
C. <i>The Case for Open Models</i>	1932
CONCLUSION	1934
APPENDIX: FOUNDATION DOCUMENTS	1936

THE HOST'S DILEMMA: STRATEGIC FORFEITURE IN PLATFORM MARKETS FOR INFORMATIONAL GOODS

Jonathan M. Barnett*

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 trade-off 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 certain 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, this Article shows that (i) contrary to standard characterizations in the legal literature, leading “open source” software projects are now primarily funded and substantially governed and staffed by corporate sponsors, and (ii) proprietary firms have formed nonprofit consortia and other cooperative arrangements and adopted “open source” licensing strategies in order to develop operating systems for the smartphone market.

In June 2008, Nokia paid \$410 million to buy out all other ownership interests in the Symbian operating system,¹ which was then the most widely used operating system in smartphone devices² worldwide.³ That would be a fairly mundane corporate acquisition if it were not for the fact that Nokia immediately transferred responsibilities for managing, developing, and distributing the operating system to the

* Associate Professor, University of Southern California Gould School of Law. I am grateful for comments from Peter Siegelman and other participants at a faculty workshop at the University of Connecticut School of Law, the Colloquium on Innovation Policy at NYU School of Law, the 2010 Annual Intellectual Property Scholars Conference at the University of California, Berkeley, School of Law, and the Law and Innovation Symposium at the University of Southern California Gould School of Law. Jinmin Chen, Daniel Fullerton, Blake Horn, Kawon Lee, and Ingrid Newquist provided excellent research assistance. This project was supported by a grant from the Southern California Innovation Project, which is funded by the Ewing Marion Kauffman Foundation. All errors are mine.

¹ J. Nicholas Hoover & Paul McDougall, *Nokia's Symbian Deal Rewrites the Smartphone Rules*, INFORMATIONWEEK, June 30, 2008, at 18, 18.

² 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.

³ Until recently, Nokia's Symbian system was the clear market leader in the smartphone operating system market, on a worldwide basis. As discussed subsequently, stiff competition, in particular from Google's Android system, has challenged its lead. For detailed information on market share, see *infra* Figure 3, p. 1918.

Symbian Foundation, a nonprofit entity.⁴ Even that transfer might be construed as a large but similarly unexceptional act of corporate largesse if it were not for the fact that Nokia, the world's leading handset maker,⁵ invited telecommunications providers, handset makers, and other firms that compete with it to serve on the Foundation's board and other governing entities.⁶ To cap off what was an exceptional sequence of events, the Foundation then spent two years clearing all third-party rights in the Symbian source code,⁷ which it made publicly available in February 2010 without charge under an "open source" license.⁸ 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 — some of their most valuable innovations to all interested parties, including customers and rivals. Examples include some of the industry's most important innovations: to name just a few, AT&T's forfeiture of transistor technology in the 1950s,⁹ Xerox's forfeiture of Ethernet local area network technology in 1979,¹⁰ and Intel's release of the Universal Serial Bus (USB) standard in 1995.¹¹ Dominant firms have developed some of the most fundamental building blocks of the digital economy at great cost and then have given away or distributed those innovations at a nominal or below-market fee, often accompanied by complementary support services and tools.

The substantial incidence and magnitude of giveaway practices in certain ICT markets challenge conventional assumptions that firms will always elect to exert maximal legal and technological control over intellectual assets, subject solely to enforcement costs.¹² Even — or

⁴ Hoover & McDougall, *supra* note 1, at 18.

⁵ Press Release, Gartner, Inc., Gartner Says Worldwide Mobile Phone Sales Grew 17 Per Cent in First Quarter 2010 (May 19, 2010), <http://www.gartner.com/it/page.jsp?id=1372013>.

⁶ Hoover & McDougall, *supra* note 1, at 18.

⁷ Source code is the human-readable form of the binary code for a software program. Reverse engineering binary code requires great cost, time, and labor.

⁸ See Darryl K. Taft, *Symbian Open-Sources Smartphone Platform*, EWEEK.COM (Feb. 4, 2010), <http://www.eweek.com/c/a/Linux-and-Open-Source/Symbian-Open-Sources-Smartphone-Platform-124651>.

⁹ See Richard C. Levin, *The Semiconductor Industry*, in GOVERNMENT AND TECHNICAL PROGRESS: A CROSS-INDUSTRY ANALYSIS 9, 75 (Richard R. Nelson ed., 1982).

¹⁰ 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, at D1.

¹¹ See *Universal Serial Bus (USB)*, INTEL, <http://www.intel.com/technology/usb/index.htm> (last visited May 5, 2011).

¹² For representative arguments in support of the conventional view, see LAWRENCE LESSIG, FREE CULTURE: HOW BIG MEDIA USES TECHNOLOGY AND THE LAW TO LOCK DOWN CULTURE AND CONTROL CREATIVITY (2004), which argues that large, concentrated media

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 a firm's most valuable intellectual assets. This Article identifies 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 rational strategy whenever inducing widespread adoption independently or contractually with third parties is more costly — which, this Article argues, is a typical (but not universal) case given certain industry-specific characteristics. Competitive pressures force "tough guys" to "play nice": the market rewards firms and other entities that act generously within limits toward rivals and customers and often severely punishes firms and entities that do otherwise.

The key to understanding forfeiture as a rational and typical practice lies in the observation that forfeiture 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 exhibit network effects — that is, the platform's value is an increasing function of the number of users¹³ and uses. Network effects imply switching costs, which, as compounded by learning costs, imply that users are subject to lock-in effects once the platform has achieved scale. At that point, the intermediary (which is called the "host") appears to enjoy pricing power over users. But that common observation can be true only from a static viewpoint. So long as users anticipate lock-in effects, the host cannot 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 it does achieve scale.

interests seek to maximize control of creative content through expansion of copyright. See also Jessica Litman, *The Exclusive Right to Read*, 13 CARDOZO ARTS & ENT. L.J. 29, 48 (1994) ("Of course, copyright owners would prefer to maximize their own control over the works they produce."); Jonathan Zittrain, *Normative Principles for Evaluating Free and Proprietary Software*, 71 U. CHI. L. REV. 265, 285 (2004) (arguing that business interests lobby for maximum legal protections for intellectual assets and resist "nonproprietary production models").

¹³ In this Article, "users" refers both to end users, who may be individuals or corporate entities, and intermediate users, who are usually corporate entities. In information technology markets, developer firms (often known as "independent software vendors" or "ISVs") are an especially important class of intermediate users and therefore play a prominent role in the subsequent discussion. Where appropriate, this Article sometimes refers specifically to end users or intermediate users.

This double commitment problem yields forfeiture as both a typical and a 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 sends a signal of confidence that encourages user adoption. Second, as this Article examines in extensive 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 holdup.¹⁴ 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 with 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. Financing requires regulating access to some portion of the consumption bundle constituted by the platform and complementary goods and services. This inherent trade-off between forfeiting and controlling access yields a testable 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).¹⁵ The

¹⁴ Other scholars have discussed, at various levels of specificity, commitment devices by which firms may provide assurance against user lock-in. For theoretical contributions in the economic literature, see *infra* note 83. For applications in the economic and management literature to platform technology markets, see sources cited *supra* note 12. For applications in the economic and management literature to the open development of certain software applications (with emphasis on sophisticated end users), see Michael Schwarz & Yuri Takhteyev, *Half a Century of Public Software Institutions: Open Source as a Solution to Hold-Up Problem*, 12 J. PUB. ECON. THEORY 609, 616–17 (2010). On technology firms' use of scientific publications as a commitment device, see Oren Bar-Gill & Gideon Parchomovsky, Essay, *The Value of Giving Away Secrets*, 89 VA. L. REV. 1857, 1858–61 (2003), which is discussed further subsequently. See *infra* note 40. This Article extends these prior contributions in three principal respects: it shows (i) how commitment concerns account for the widespread use of forfeiture practices in a variety of infrastructural technology settings, irrespective of commercial or noncommercial motivations; (ii) how firms strategically use nonprofit entities, permissive licenses, and "community" norms in order to address intermediate and end users' lock-in concerns in horizontal technology markets (in particular, in order to promote the adoption of operating systems in the enterprise-computing and smart-phone markets), subject to the constraints imposed by the cost recovery imperative; and (iii) how the strategic purposes of forfeiture practices cast doubt on the standard normative preference for open over closed innovation models.

¹⁵ In a related line of inquiry, management and some economic scholars have recently applied an "openness versus appropriability" trade-off to assess the extent to which platform holders can regulate access in order to provide assurance against user lock-in while still maintaining a positive revenue stream. See generally Joel West, *How Open Is Open Enough? Melding Proprietary and Open Source Platform Strategies*, 32 RES. POL'Y 1259 (2003). For further discussion, see Thomas

greater the control the host forfeits, the stronger its ability to induce user adoption, but the weaker its ability to capture revenues that at least cover development costs. Conversely, the lesser the control the host forfeits, 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 because the former limit user adoption (which would diminish the platform's value) and the latter limit revenue accrual (which would make it difficult for the platform to 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, yielding 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 reveals 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. This Article starts by reviewing the organizational forms used historically to develop and distribute operating systems for personal- and enterprise-computing devices. Consistent with theoretical expectations, these markets tend to rely on a mix of open and closed elements in order to induce platform adoption within the constraints of business prudence. The Article then studies in greater detail the organizational forms used in two contemporary operating system markets. First, this Article shows that hybrid structures characterize leading “open source” software (OSS) projects,¹⁶ which are governed by nonprofit foundations that are entirely funded and substantially managed and staffed by commercial sponsors. These findings depart sharply from the conventional characterization in the legal literature of open source projects as spontaneously organized communities of intrinsically motivated volun-

R. Eisenmann et al., *Opening Platforms: How, When and Why?*, in *PLATFORMS, MARKETS AND INNOVATION* 131, 131–62 (Annabelle Gawer ed., 2009). For empirical applications, see Kevin Boudreau, *Open Platform Strategies and Innovation: Granting Access vs. Devolving Control*, 56 *MGMT. SCI.* 1849, 1854–56 (2010), with application to handheld computing; and Annabelle Gawer & Rebecca Henderson, *Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel*, 16 *J. ECON. & MGMT. STRATEGY* 1, 9–25 (2007), with application to Intel's Architecture Lab.

¹⁶ Open source software projects release the software's human-readable source code at no fee and with few restrictions on use and distribution. By contrast, proprietary software is released in binary object-code form for a fee and under strict contractual restrictions on use and distribution.

teer programmers.¹⁷ Second, this Article examines the rapidly evolving smartphone market, where leading handset makers, telecommunications providers, internet search companies, and semiconductor chip makers have sought to elicit developer adoption of operating systems by forming nonprofits and other consortia to develop those systems and by offering those systems at no cost under licenses that disclose the underlying source code. These findings depart sharply from the conventional view in the legal literature that for-profit firms consistently seek to use all available legal and other means to limit access by rivals and consumers to technological assets.¹⁸ In both the smartphone and open source software markets, controlled generosity follows from economic self-interest: implicit or explicit consortia of commercial firms open up access in order to commit against host opportunism and 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 and, even more clearly, rebuts any inherent association of open structures with noncommercial entities and closed structures with commercial entities. Organizational convergence derives from the fundamental trade-off that confronts any platform holder. For-profit firms adopt open structures in order to commit against host opportunism, while (ostensibly) nonprofit 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 standard normative presumption — widely endorsed (or simply assumed) in the legal and broader policy literature on OSS and related innovation environments — that the public interest inherently favors the adoption of open over closed structures.¹⁹ If open and closed structures (and all intermediate variants) simply reflect strategic approaches to the underlying trade-off between controlling host opportunism and enabling cost recovery, then

¹⁷ See, e.g., YOCHAI BENKLER, *THE WEALTH OF NETWORKS: HOW SOCIAL PRODUCTION TRANSFORMS MARKETS AND FREEDOM* 60, 63–64 (2006); Yochai Benkler, *Coase's Penguin, or, Linux and The Nature of the Firm*, 112 YALE L.J. 369, 381–84 (2002); Yochai Benkler, Essay, *Sharing Nicely: On Shareable Goods and the Emergence of Sharing as a Modality of Economic Production*, 114 YALE L.J. 273, 334–36 (2004) [hereinafter Benkler, *Sharing Nicely*]; James Boyle, *The Second Enclosure Movement and the Construction of the Public Domain*, LAW & CONTEMP. PROBS., Winter–Spring 2003, at 33, 44–45. The string of foregoing sources in this footnote is a selective, not a comprehensive, list of legal contributions that advance this characterization of open source software. These works build upon views expressed by the most well-known work on the subject in popular commentary. See generally ERIC S. RAYMOND, *THE CATHEDRAL AND THE BAZAAR: MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY* (1999).

¹⁸ See sources cited *supra* note 12.

¹⁹ See *infra* pp. 1926–27.

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 competing providers offer in the strategic pursuit of market share. Moreover, there is no assurance that open structures even promote consumer welfare. That surprising possibility arises whenever forfeiture exerts entry-deterrent effects that protect dominant firms against potential competitors. A host that forfeits its platform technology compels stand-alone platform providers to exit 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.

The organization of this Article is as follows. Part I describes illustrative forfeiture practices in ICT markets. Part II describes the host's dilemma and possible solutions through contract, integration, and forfeiture. Part III shows 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. Part IV discusses how the strategic motivations behind forfeiture practices complicate policy preferences for open over closed innovation models.

I. VOLUNTARY FORFEITURE: A TYPICAL PRACTICE

It is often naturally assumed that firms seek to exercise maximal control over technological assets and, even more so, over the most valuable technological assets. This assumption arguably motivates the unusual amount of scrutiny dedicated by legal, economic, and management scholars to the apparent anomaly represented by OSS projects²⁰: individual programmers contribute code without compensation, and project management then releases the accumulated source code at no fee and with few restrictions on use. But even casual scrutiny shows that free distribution is neither a historical anomaly nor a peculiarity of noncommercial environments. Some contemporary examples are familiar to users who stand at the end of the ICT supply

²⁰ For an overview of the literature, see Stephen M. Maurer & Suzanne Scotchmer, *Open Source Software: The New Intellectual Property Paradigm* (Nat'l Bureau of Econ. Research, Working Paper No. 12148, 2006), available at <http://www.nber.org/papers/w12148.pdf>. For a readable (and nuanced) book-length overview of the OSS market, see STEVEN WEBER, *THE SUCCESS OF OPEN SOURCE* (2004).

chain: Google and Microsoft provide access to their search engines at no cost, Facebook and Twitter provide access to social networking applications at no cost, Adobe releases basic versions of its Reader software at no cost, Microsoft releases its browser application for download at no cost, and so on. As Table 1 shows, these recent acts of generosity by technology firms simply extend a pattern of behavior in the ICT industry that was present at the industry's inception. Contrary to natural intuitions, some of the most dominant firms and other entities in the ICT sector have regularly given away some of their most valuable technologies to the general public, which inherently includes actual and potential rivals and customers. The following discussion examines in further detail two of the most notable voluntary forfeitures in ICT history.

A. Bell Labs: Open Licensing

The licensing practices of Bell Labs, the research laboratory founded by AT&T in 1925 (and eventually owned by Lucent Technologies, an AT&T spin-off entity launched in 1996),²¹ probably constitute the single greatest act of corporate generosity in the technology sector. Bell Labs is credited with developing some of the twentieth century's leading inventions, including the transistor, the UNIX operating system, and key technologies behind cellular mobile communications.²² Until its judicially ordered breakup in 1984, AT&T made its key innovation — the transistor — and other technologies relating to communications services available at nominal royalties subject to a cross-licensing obligation.²³ Although these policies were mandated under a 1956 consent decree settling federal antitrust litigation, AT&T had instituted roughly the same policies with respect to the transistor and certain electrical equipment used in telephone services prior to the consent decree.²⁴ Despite its strong patent position, AT&T had

²¹ See *Alcatel-Lucent History*, ALCATEL-LUCENT, <http://www.alcatel-lucent.com/wps/portal/AboutUs/Overview> (follow "Alcatel-Lucent History" hyperlink) (last visited May 5, 2011). Presently, the parent company of Bell Labs is Alcatel-Lucent, the entity created when Alcatel merged with Lucent Technologies in 2006. *Id.*

²² See NATHAN J. MULLER, *DESKTOP ENCYCLOPEDIA OF TELECOMMUNICATIONS* 79–80 (3d ed. 2002).

²³ See Levin, *supra* note 9, at 75; see also Deepak Somaya & David J. Teece, *Patents, Licensing, and Entrepreneurship: Effectuating Innovation in Multi-Invention Contexts*, in *TECHNOLOGICAL KNOW-HOW, ORGANIZATIONAL CAPABILITIES, AND STRATEGIC MANAGEMENT* 287, 298 (David J. Teece ed., 2008) (noting that, until its breakup in 1984, AT&T's policy was "to openly license its IP to everyone for minimal fees"). AT&T's licensing policies with respect to the UNIX operating system are discussed in further detail subsequently. See section III.A.1, pp. 1891–92.

²⁴ Levin, *supra* note 9, at 75–76.

TABLE 1: SELECTED FORFEITURE ACTIONS IN ICT MARKETS²⁵

<i>Date</i>	<i>Firm(s)</i>	<i>Forfeiture Action</i>
1940s–1984	Bell Labs (AT&T)	Licensed transistor and related technologies at nominal royalty, subject to cross-licensing obligation. Formalized licensing practices in 1956 consent decree.
1980	Xerox, Intel, Digital Equipment (DEC)	Licensed Ethernet local area network technology at nominal fee.
1980s–present	Microsoft	Disclosed Windows application programming interfaces to independent software developers.
1995	Intel	Released USB interface technology through USB Implementers Forum (USB-IF), a nonprofit trade organization.
1998	IBM, Intel, Ericsson, Nokia, Toshiba	Founded the Bluetooth Special Interest Group (SIG), a nonprofit trade association that publishes Bluetooth technical specifications and oversees licensing of Bluetooth marks and technologies (for wireless communication devices).

²⁵ On Bell Labs, see *infra* pp. 1870–72. On Xerox, see HILTZIK, *supra* note 10, at 363–64 (1999); and Markoff, *supra* note 10, at D1. On Microsoft, see MARCO IANSITI & ROY LEVIEN, *THE KEYSTONE ADVANTAGE: WHAT THE NEW DYNAMICS OF BUSINESS ECOSYSTEMS MEAN FOR STRATEGY, INNOVATION, AND SUSTAINABILITY* 56 fig.3-6 (2004); and *infra* pp. 1872–74. On Intel's release of the USB interface, see *Universal Serial Bus (USB)*, *supra* note 11. On the release of Bluetooth technology, see *About the Bluetooth SIG: Overview*, BLUETOOTH.ORG, https://www.bluetooth.org/About/bluetooth_sig.htm (last visited May 5, 2011). On Netscape, see Janet Kornblum, *Netscape Sets Source Code Free*, CNET NEWS (Mar. 31, 1998, 12:10 PM), <http://news.cnet.com/2100-1001-209666.html>. On IBM's investment in Linux and other related actions, see Joe Wilcox, *IBM to Spend \$1 Billion on Linux in 2001*, CNET NEWS (Dec. 12, 2000, 8:50 AM), <http://news.cnet.com/2100-1001-249750.html>. On IBM's donation to Eclipse, see David Gallardo, *Getting Started with the Eclipse Platform*, IBM (Nov. 1, 2002), <http://www.ibm.com/developerworks/library/os-ecov/index.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>; and, for further references, see *infra* note 156. On Sun's release of the Solaris source code, see Michael Singer, *Sun Cracks Open Solaris*, INTERNETNEWS.COM (Jan. 25, 2005), <http://www.internetnews.com/dev-news/article.php/3463621>. On Sun's release of the Java source code, see Ryan Paul, *Sun to Release Java Micro Edition Source Code*, ARS TECHNICA, <http://arstechnica.com/old/content/2006/08/7514.ars> (last visited May 5, 2011). On Sun's open practices more generally, 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).

<i>Date</i>	<i>Firm(s)</i>	<i>Forfeiture Action</i>
1999	Netscape	Released source code for its browser technology.
2000	IBM	Announced \$1 billion commitment to (open source) Linux operating system.
2005	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 and established community advisory council for governance.
2006	Sun Microsystems	Released source code for Java programming language, subject to limited contractual restrictions.

adopted a policy of actively disseminating its technologies and even assisting third parties in using them.²⁶ A former AT&T executive stated this policy 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 early 1950's where we effectively told all we knew about transistor technology.²⁷

Remarkably, this statement tracks almost exactly an Intel executive's statement concerning his firm's open licensing of the USB standard²⁸ several decades later: "We developed the [USB] code And we also made it available to anybody in the industry."²⁹ As this Article shows, 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 using open models for developing software applications. As is widely observed, Microsoft's success rests in part on its release of application programming interfaces³⁰ (APIs) for the Windows operat-

²⁶ See JOHN E. TILTON, *INTERNATIONAL DIFFUSION OF TECHNOLOGY: THE CASE OF SEMICONDUCTORS* 74 (1971).

²⁷ Morgan Sparks, *25 Years of Transistors*, 50 *BELL LABORATORIES REC.* 342, 343-44 (1972).

²⁸ The USB standard is a peripheral interface that enables communication between a computer and external devices such as printers, keyboards, and flash memory disks.

²⁹ Gawer & Henderson, *supra* note 15, at 19 (alteration in original) (quoting Interview with Jim Pappas, Dir. of Platform Initiatives, Desktop Prods. Grp., Intel Corp. (Aug. 4, 1998)).

³⁰ An application programming interface is a language and message format used by an application program to communicate with the operating system or other application programs.

ing system to outside developers³¹ and its extensive efforts to construct a product architecture and communications infrastructure that facilitate third-party development of complementary applications.³² For access to Windows APIs and related technical information, Microsoft has often assessed an implicit negative fee taking into account the software development tool kits and support services it provides.³³ Moreover, Microsoft has incurred significant costs — both direct programming costs and indirect costs in the form of product quality³⁴ — in order to make its APIs “backward compatible” across Windows versions,³⁵ thus allowing existing applications to operate on newer versions of Windows. This is not to say that Microsoft does not restrict 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 have feasibly and legally exerted control.³⁶ The fact that its less successful competitors (in retrospect, unwisely) chose not to do so — for example, Apple, which pursued a largely closed development strategy

³¹ See IANSITI & LEVIEN, *supra* note 25, at 30, 88–89; Shane Greenstein, *Open Platform Development and the Commercial Internet*, in PLATFORMS, MARKET AND INNOVATION 219, 223 (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, 865 (2009); William H. Page, *Mandatory Contracting Remedies in the American and European Microsoft Cases*, 75 ANTITRUST L.J. 787, 800 (2009). Microsoft’s initial disclosure of interface specifications appears to have been done at the demand of IBM, its client, which sought to elicit adoption of the new operating system by third-party developers. See Pamela Samuelson, *IBM’s Pragmatic Embrace of Open Source*, COMM. ACM, Oct. 2006, at 21, 22. Of course, the continued disclosure of the API specifications for several decades thereafter is a purely voluntary decision by Microsoft, which is not under any legal or contractual obligation to disclose them.

³² See MICHAEL A. CUSUMANO & RICHARD W. SELBY, MICROSOFT SECRETS: HOW THE WORLD’S MOST POWERFUL SOFTWARE COMPANY CREATES TECHNOLOGY, SHAPES MARKETS, AND MANAGES PEOPLE 168–69, 236–38 (1995); IANSITI & LEVIEN, *supra* note 25, at 30, 88; Greenstein, *supra* note 31, at 223. Tellingly, Intel — Microsoft’s partner in crime from the perspective of some antitrust authorities — has pursued the same cooperative strategy with respect to some of its most fundamental technologies. See Gawer & Henderson, *supra* note 15, at 19.

³³ See DAVID S. EVANS ET AL., INVISIBLE ENGINES: HOW SOFTWARE PLATFORMS DRIVE INNOVATION AND TRANSFORM INDUSTRIES 66 (2006).

³⁴ Backward compatibility can reduce product quality to the extent that it limits freedom of development in newer versions.

³⁵ See IANSITI & LEVIEN, *supra* note 25, at 167; Greenstein, *supra* note 31, at 223.

³⁶ More recently, Microsoft has undertaken further commitments to disclose its technology (including portions of the Windows source code) to outside parties, including the Shared Source Initiative, see *Shared Source Initiative*, MICROSOFT, <http://www.microsoft.com/resources/sharedsource/default.aspx> (last visited May 5, 2011), the Open Specification Promise, see *Microsoft Open Specification Promise*, MICROSOFT, <http://www.microsoft.com/interop/osp/default.aspx> (last updated Feb. 1, 2007), and the Interoperability Principles, see *Interoperability Principles*, MICROSOFT (Feb. 21, 2008), <http://www.microsoft.com/interop/principles/default.aspx>. This Article omits discussion of 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.

in competing with Windows and struggled to elicit widespread adoption of its Macintosh product line — 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? Some scholars have referred principally to noneconomic factors such as altruism, ideology, and intrinsic interest in order to account for individual contributions of knowledge assets in ostensibly noncommercial settings such as OSS.³⁷ But those factors do not plausibly apply to commercial entities obligated by legal and business pressures to maximize owners' profits. Existing nonaltruistic explanations include the following: (i) firms are compelled to forfeit knowledge assets that cannot be protected at a reasonable cost, (ii) firms wish to preempt patenting by competitors,³⁸ (iii) firms seek to recruit researchers who wish to accumulate reputational capital in the scientific community, and (iv) firms wish to build a large installed base.³⁹ These explanations have merit, but they are generally case-specific explanations, have difficulty accounting for the forfeiture of especially valuable technological assets, or, in the last case, are incomplete for reasons described below. This Part applies a simple rationale: host entities forfeit platform assets in order to commit credibly against expropriating users' investments in those assets.⁴⁰ As discussed in

³⁷ See sources cited *supra* note 17.

³⁸ See Douglas Lichtman et al., *Strategic Disclosure in the Patent System*, 53 VAND. L. REV. 2175 (2000); 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).

³⁹ 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. INDUS. ECON. 155 (1994).

⁴⁰ For prior contributions in the management and economic literature that have explored the extent to which platform holders open up access in order to address users' lock-in concerns, see *supra* note 14. The following discussion provides a somewhat narrower formulation of that line of argument (in part because the historical evidence seems to show that (transiently) dominant repeat-player firms' lock-in capacities are more tenuous than is often stated to be the case). In a related context, Professors Oren Bar-Gill and Gideon 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 Bar-Gill & Parchomovsky, *supra* note 14. This Article's analysis is consistent with (and empirical evidence further confirms) that argument; however, the controlled forfeiture practices on which this Article focuses are more elaborate than merely abandoning knowledge through simple disclosure. Three considerations account for this increased complexity: (i) credibly committing against expropriating user investments requires sequestering knowledge in an entity over which the host cannot exercise or reclaim control; (ii) simple abandonment in the absence of a centralized coordinating agent can fail to elicit user investments because users fear volatile technological standards that will endanger the value of their investments; and (iii), as this Article emphasizes in particular, any forfeiture

greater detail below, host entities employ a rich set of forfeiture strategies in order to promote that objective.

A. Some Economics of Platform Markets

Any ICT platform — which may be constituted by hardware, software, or an operating system⁴¹ — 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. Network effects amplify transaction cost savings plus associated trading gains. That is, any user's gains are an increasing function of the number of other users of, or uses for, the platform technology.⁴² As shown in Figure 1, 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: developer user–end user, end user–end user, and developer user–developer user.⁴³ 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 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).⁴⁴ Microsoft Windows, Sony PlayStation,

action requires regulating access to some complementary good in order to generate revenues to cover (at least) the host's development and other costs. For further discussion of points (i) and (ii), see section II.C.3.b, pp. 1888–89; with respect to point (iii), see Part III, pp. 1890–1925.

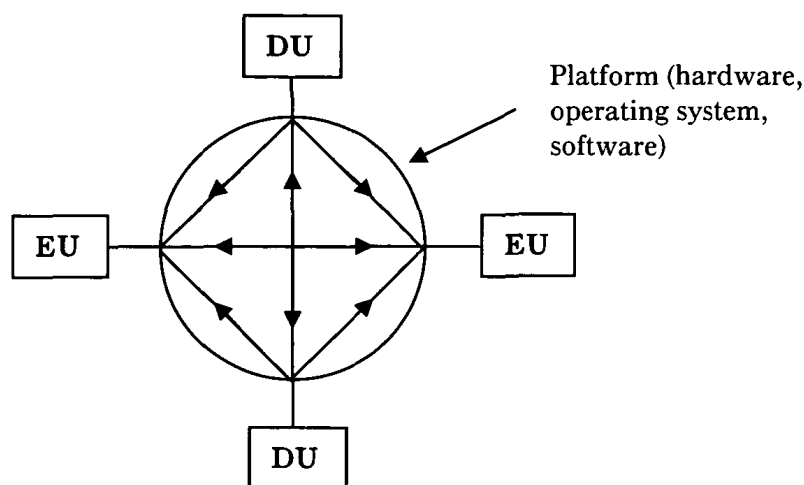
⁴¹ Slightly more and less expansive definitions of this constituent set are sometimes used (for example, “operating system” can be construed as a type of software, while the software category can be subdivided to include middleware applications). For a fuller explanation in nontechnical terms, see MARC H. MEYER & ALVIN P. LEHNERD, *THE POWER OF PRODUCT PLATFORMS* (1997).

⁴² These attributes correspond, 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).

⁴³ For simplicity, the Figure does not reflect that other intermediaries, such as systems integrators, will typically occupy points on the transaction path between end users and developer users (as is the case with 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.

⁴⁴ For simplicity, assume that (i) neither user group suffers from congestion costs, (ii) both end users and developer users enjoy constant increasing returns from additional developer users or end users, respectively, and (iii) end users and developer users value, respectively, all types of developer users and end users equally (or equivalently, end users and developer users are homogenous).

FIGURE 1: TRANSACTION PATHS IN ICT PLATFORM MARKETS



and the Apple iPhone (or, to take an even more fundamental example, the abundance of communications and other devices enabled by Bell Labs' invention of the transistor) confer transactional gains by (i) connecting developer users (for example, video game developers) to end users (for example, video game players) and (ii) connecting end users to other end users (for example, iPhone users) or developer users to other developer users (for example, developers of Windows-compatible software applications, who effectively deliver an applications suite to consumers). 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 and 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 erode the platform's value or stunt adoption altogether. These feedback effects are 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 toward — or, in a period of decline, away from — the same platform. Even the most dominant platform therefore inherently occupies

a precarious position, as 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.⁴⁵ 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 1981⁴⁶ but was rapidly overwhelmed by clones and exited from the market entirely in a sale to Lenovo in 2005;⁴⁷ Palm distributed the first successful handheld computing device, achieving a 70% U.S. market share by 1997,⁴⁸ but was overwhelmed by RIM's BlackBerry device in the early 2000s and was sold in a distressed transaction to Hewlett-Packard in 2010;⁴⁹ Netscape was the prevailing internet browser, achieving an 80% market share in 1995,⁵⁰ but eventually lost the leadership position to Microsoft's Internet Explorer and had a nominal market share by the early 2000s.⁵¹ The best example may be America Online (AOL): its apparent dominance of the internet service portal market was so powerful that, in 2001, it could merge with and effectively acquire media conglomerate Time Warner;⁵² by 2009, however, the combined entity had spun off the declining AOL, which ac-

⁴⁵ Even Microsoft's dominance in desktop computing is vulnerable to competition in adjacent markets for partially substitutable computing devices. On May 26, 2010, Apple surpassed Microsoft in stock market capitalization, reflecting the partial displacement of the personal computer market by the smartphone and tablet devices markets (where Apple's products are among the market leaders and Windows-based devices are laggards). See Connie Guglielmo & Dina Bass, *Apple Overtakes Microsoft in Market Capitalization*, BLOOMBERG BUSINESSWEEK (May 26, 2010), <http://www.businessweek.com/news/2010-05-26/apple-overtakes-microsoft-in-market-capitalization-update3-.html>. For further information on competitive conditions in the smartphone market, see *infra* pp. 1917–20.

⁴⁶ See *IBM Personal Computer*, IBM, http://www-03.ibm.com/ibm/history/exhibits/pc/pc_1.html (last visited May 5, 2011).

⁴⁷ See *Lenovo of China Completes Purchase of I.B.M.'s PC Unit*, N.Y. TIMES, May 2, 2005, at C5.

⁴⁸ Press Release, Palm, Inc., Dataquest and PC Data Research Studies Place U.S. Robotics PalmPilot at Lead of Thriving Handheld Computer Market (May 12, 1997), <http://investor.palm.com/releasedetail.cfm?releaseid=336348>.

⁴⁹ See Associated Press, *HP to Acquire Palm for About \$1 Billion in Cash*, FOXNEWS.COM (Apr. 28, 2010), <http://www.foxnews.com/scitech/2010/04/28/hp-acquire-palm-nearly-billion-cash>.

⁵⁰ Jared Sandberg, *Sun and Netscape Are Forming Alliance Against Microsoft on Internet Standard*, WALL ST. J., Dec. 4, 1995, at B8.

⁵¹ See Susan Stellan, *Browser Battle Winds Down*, N.Y. TIMES, Sept. 9, 2002, at C6. As described subsequently, the Netscape code later formed the basis for the open source Firefox browser, which is managed by the Mozilla Foundation. See *infra* note 84. Reflecting the volatility of platform dominance, Firefox had a 22.97% market share as of September 2010. See *Browser Market Share*, NETMARKETSHARE (Sept. 2010), <http://marketshare.hitslink.com/firefox-market-share.aspx?qprid=0&sample=28&qptimeframe=M&qpsp=140>; see also *infra* p. 1900.

⁵² See Stephen Labaton, *F.C.C. Approves AOL-Time Warner Deal, with Conditions*, N.Y. TIMES, Jan. 12, 2001, at C1.

counted for a negligible percentage of the U.S. “core search” market.⁵³ The appearance of platform dominance can often be illusory — misleading even the most sophisticated academic observers,⁵⁴ antitrust judges, and market investors — and the movement from leader to laggard can often be swift and brutal.

B. *The Double Commitment Problem*

To elicit user adoption, a platform holder must overcome two obstacles. First, it must persuade initial users that the platform will scale and therefore deliver the network effects that give it value over competing platforms (or the alternative of nonmediated communication). Second, it must persuade users that, even after the platform achieves scale and delivers value in the form of network effects, the platform holder will not regulate access in order to expropriate that value from users. Each of these commitment problems is addressed in turn.

1. *The Intertemporal Dilemma.* — 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 other users adopt it in sufficient numbers. For example, end users are reluctant to adopt a platform until a large mass of other end users or applications (or both) has materialized, which in turn means that developer users 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.

But this “waiting game” 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 price⁵⁵ (which the host may pay in kind in the form of technical assistance or other support). That strategy explains risky gambles such as JVC’s decision in the 1970s to widely license the VHS technology for videocassette recorders, which prevailed over the competing Betamax technology that Sony kept to itself,⁵⁶ or Xerox’s decision in 1979 to li-

⁵³ See Jonathan E. Skillings, *Time Warner to Spin Off AOL*, CNET NEWS (May 28, 2009, 5:18 AM), http://news.cnet.com/8301-1023_3-10250944-93.html.

⁵⁴ Commentators’ laments of “unbeatable” network monopolies often prove to be short-sighted. For example, Professors Steven Salop and R. Craig Romaine cited the Palm Pilot as an exemplary case of a network good that attracts market adoption through positive feedback effects, which are then argued to be an entry barrier that is difficult to overcome. See Steven C. Salop & R. Craig Romaine, *Preserving Monopoly: Economic Analysis, Legal Standards, and Microsoft*, 7 GEO. MASON L. REV. 617, 621 (1999). With hindsight, the Palm Pilot demonstrates the ease with which market dominance can be lost. See *supra* p. 1877.

⁵⁵ See Nicholas Economides, *Network Externalities, Complementarities, and Invitations to Enter*, 12 EUR. J. POL. ECON. 211 (1996).

⁵⁶ See Michael A. Cusumano et al., *Strategic Maneuvering and Mass-Market Dynamics: The Triumph of VHS over Beta*, 66 BUS. HIST. REV. 51 (1992) (discussing dynamic standardization in

cense its Ethernet local area network technology at a nominal fee, which made Ethernet the global networking standard.⁵⁷ Each of these costly transfers from the host to early adopters functions as a bond posted by the host, which stands to suffer a financial penalty if the platform fails to scale as expected. That is, the host “burns money” at an initial stage in order to signal to early adopters its confidence that its platform will achieve sufficient scale to recoup those costs at a subsequent stage. This bonding strategy can provide some assistance in eliciting user adoption of a platform technology. However, as section 2 shows, it is incomplete in a fundamental respect.

2. *The Host's Dilemma.* — Even if the host can post a bond by which it persuades users that the platform will achieve sufficient scale, the host still will not have overcome all obstacles to user adoption. The host must still persuade users that it will leave them with a net gain after the platform has achieved scale. Users are 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 what is called the host's dilemma.

(a) *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 assumption is true in the case of end users, and to a substantially greater extent, it is also true in the case of developer users, who must invest substantial sums in developing, marketing, and supporting applications for use on the platform. If this characteristic applies 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 host's costs or improve certain platform features but reduce the value of users' investments in the existing platform (for example, an upgrade that makes a developer's applications incompatible with the platform), and integrating forward into a developer user's market. In

the VCR industry and the minimal first-mover advantages for products characterized by bandwagon effects).

⁵⁷ See HILTZIK, *supra* note 10, at 363–64. The former gamble resulted in commercial success for the host; the latter did not. That Xerox failed to capitalize on its success (which the company had hoped would promote the sale of complementary product lines where it had a competitive advantage) is the type of outcome to which a host can refer in order to make its second credible commitment (discussed immediately below) — namely, that the host will have limited ability to exercise pricing power over early adopters, who will be free to purchase from competing providers.

the last case, the prospect of a well-capitalized and well-branded platform holder competing with a much smaller developer firm directly threatens the developer's existing profit stream.⁵⁸ Not coincidentally, all of these accusations are routinely leveled both formally and informally against the most salient platform holders — for example, Microsoft and Intel — in informational goods markets.

Transaction cost economics (in particular, as pioneered by Nobel Prize winner Professor Oliver Williamson⁵⁹) provides a concise terminology for describing this state of affairs. Users make *ex ante* investments that are “specific” to the platform (that is, have no use outside the platform), which implies an *ex post* expropriation opportunity for the host, who then can regulate access in order 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 result is paradoxical. Precisely at the point where a platform has achieved the highest levels of user adoption, given the expropriation threat (and assuming the lack of competing platforms), users derive both the greatest potential benefit from the platform (since users' gains from network effects are at their highest) and the lowest actual benefit (since 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, the user 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, the host cannot induce platform

⁵⁸ On forward integration by platform holders and the competitive effect on applications developers, see Annabelle Gawer, *Platform Dynamics and Strategies: From Products to Services*, in *PLATFORMS, MARKETS AND INNOVATION* 45, 57 (Annabelle Gawer ed., 2009). For a discussion of the same dynamic in the specific context of Intel, see Gawer & Henderson, *supra* note 15, at 9–25.

⁵⁹ For the leading work, see generally OLIVER E. WILLIAMSON, *MARKETS AND HIERARCHIES: ANALYSIS AND ANTITRUST IMPLICATIONS* (1975). For a shorter presentation of this approach, see generally Oliver E. Williamson, *The Vertical Integration of Production: Market Failure Considerations*, 61 *AM. ECON. REV.* 112 (1971).

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 cannot persuade users that it will not expropriate value from them after scale has been achieved. As a result, the platform is never adopted at all (or, less dramatically, is underadopted).

(b) *The Complex Case*. — In defense of the conventional view, one might object that the perfect foresight assumption — that is, the fully rational user — is unrealistic. Platforms *are* in fact adopted in ICT markets. Consistent with the argument set forth above, that fact suggests that users fail to anticipate expropriation opportunities and that the host deceives 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.⁶⁰ 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 misleads only 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.⁶¹

I 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.⁶² User myopia would allow the host to

⁶⁰ Even that claim 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.

⁶¹ There is one contingency under which the "myopic user" objection carries some (albeit highly limited) weight. The host could make side payments to far-sighted developers in order to split the gains from luring myopic end users. Even setting aside legal constraints on that type of behavior, explicit side payments seem implausible given coordination costs, assuming either a diffuse population of software developers in real-world markets (as is the case) or a competitive population of software developers, who would compete by diverting side payments to end users or intermediate users.

⁶² In its antitrust allegation that Microsoft had induced developers to write programs for the Windows-specific Java development tool (rather than for the cross-platform development tool offered by Sun), the federal government was forced to rely precisely on the implausible claim that

elicit adoption to the extent that users fail to anticipate the host's future opportunistic behavior. But this change in assumptions would still make no difference. Unless it is further assumed (even more implausibly) that users are both perfectly myopic and perfectly forgetful, the host would have only a single opportunity to engage in opportunistic behavior. That solution is insufficient in the case of any repeat-player host that seeks to maximize long-term profits through repeated adoption of platform extensions. Microsoft cannot make any more money by selling the Windows operating system again.⁶³ Microsoft must convince users to buy the latest version of Windows, which explains in part why it has released twelve versions of the Windows operating system for desktop computing from November 1985 (when it released Windows 1.0) to the present (the latest release being Windows 7 in October 2009).⁶⁴

At each release point, the host must confront the host's dilemma: 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 upgrade to an operating system typically reach several billions of dollars: Windows 7 cost approximately \$1.5 billion,⁶⁵ Windows Vista (released 2007) cost approximately \$6 billion,⁶⁶ and Apple's Mac OS X (released 2001) cost approximately \$1 billion.⁶⁷ Eliciting user adoption of any current platform extension, and thereby recovering these substantial investments, rests on maintaining a past record of good behavior. Given the sums at stake, any past failure to build such a record exposes the host to substantial or exorbitant financial penalties in the

professional developers were unaware of the differences between these products. See *United States v. Microsoft Corp.*, 253 F.3d 34, 76–77 (D.C. Cir. 2001). On this claim (and the lack of any evidence offered in support), see David McGowan, *Has Java Changed Anything? The Sound and Fury of Innovation Litigation*, 87 MINN. L. REV. 2039, 2044–48 (2003).

⁶³ This assertion assumes (as is generally the case) that Microsoft sells Windows on a one-time basis rather than leasing access to it on a continuous basis. Even if Microsoft did offer a leasing option, the company would still be compelled to release (and induce adoption of) new versions of Windows in order to avoid rapid technological obsolescence.

⁶⁴ See *A History of Windows*, MICROSOFT, <http://windows.microsoft.com/en-US/windows/history> (last visited May 5, 2011).

⁶⁵ See David B. Yoffie & Renee Kim, *Apple Inc. in 2010*, at 6 (Harvard Bus. Sch., Case No. 9-710-467, 2010).

⁶⁶ Marius Oiaga, *Vista — A \$6 Billion Dollars Operating System*, SOFTPEDIA (Jan. 10, 2007, 12:03 AM), <http://news.softpedia.com/news/Vista-a-6-Billion-Dollars-Operating-System-44096.shtml>. Other authors cite even higher cost estimates. See Dean Takahashi, *Why Vista Might Be the Last of Its Kind*, SEATTLE TIMES, Dec. 4, 2006, available at http://seattletimes.nwsourc.com/html/businesstechnology/2003460386_btview04.html (estimating total development cost of \$10 billion, assuming a cost of \$200,000 per employee and based on *BusinessWeek* estimate of 10,000 Microsoft employees dedicated to the project over five-year period).

⁶⁷ Yoffie & Kim, *supra* note 65, at 7. All costs are estimates; but note further that all costs are substantial underestimates insofar as they exclude marketing and other implementation costs.

present. The host will refrain from acting opportunistically 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 that seeks to maximize long-term revenues has little to no incentive to exploit its expropriation opportunity. User lock-in would be a virtual impossibility.

One might therefore conclude 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 in the media, in academia, and in the policymaking community that dominant holders of platform technologies — again, Microsoft and Intel — do not always seem to behave “nicely” toward existing users. Of course, even a widespread impression may be found to be mistaken upon further examination. But 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 hosts would act in this manner — and can explain this behavior without relying on implausible or “stretched” assumptions of universally myopic and amnesiac users (or, for that matter, on irrational or incompletely informed host entities). First, if the host's managers are not employed or do not expect to be employed by the host across generations of platform extensions, then managers may have short-term incentives to expropriate users' investments in any given extension even if doing so is not consistent with long-term profit maximization. This possibility implies that platform holders (and users) are victims of platform managers. This divergence of interests between managers and the firm is well documented in the management literature on “cultural” obstacles to adopting open innovation models in certain firms.⁶⁸ This divide may be further exacerbated where (i) managers are awarded in part through equity-based compensation (as is typical in technology markets),⁶⁹ and (ii) the capital markets overweight short-term user adoption relative to long-term user attrition.⁷⁰ 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 economic gain from expropriation exceeds

⁶⁸ See, e.g., Gawer & Henderson, *supra* note 15, at 18–19, 23 (discussing Intel's difficulty in persuading managers to accept that Intel would be providing proprietary technology to competitors in order to grow the larger market).

⁶⁹ See Mark C. Anderson et al., *Executive Compensation in the Information Technology Industry*, 46 MGMT. SCI. 530 (2000) (examining the common use of stock options in the information technology industry as part of executive compensation).

⁷⁰ On short-term biases in the capital markets in general, see Kevin J. Laverty, *Economic “Short-Termism”: The Debate, the Unresolved Issues, and the Implications for Management Practice and Research*, 21 ACAD. MGMT. REV. 825 (1996).

the long-term reputational penalty.⁷¹ Third, at the point at which any platform (or platform extension) is released, the host cannot 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 theoretical contingency corresponds in practical terms 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 will assign a positive likelihood that the host may expropriate in the future, in which case users will adopt only subject to a discount that reflects that nonexcludable contingency. Hence the host's dilemma persists: even if it is a repeat player, the host cannot fully commit against expropriation within these limited but typical contingencies and therefore cannot induce users to adopt the platform (or, more precisely, cannot do so without offering a discount).

C. Solutions to the Host's Dilemma

The host's dilemma implies that platforms will be substantially underadopted. Clearly that is not the case: platforms are endemic to ICT markets.⁷² 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, one must assess the relative effectiveness of three devices that address the host's commitment problem: contract, integration, and forfeiture. Together or individually, these devices enable the host to construct an organizational and transactional structure by which it can commit against future expropriation after scale has been achieved.

1. *Contract.* — The host can attempt to write a contract that binds it against opportunistic behavior. This solution is meaningful but imperfect for several reasons: (i) the user's ability to enforce the contract is limited by the host's life, solvency, and legally attachable assets; (ii)

⁷¹ I am referring to a standard model of indefinite or infinite repeat play where a single defection results in irrevocable ejection from the game and the loss of all future cooperative gains. For example, in the most well-known formulation, the "tit-for-tat" game, a successful player elects *cooperate* in the initial round of an iterated sequence and in each round thereafter but then reverts irrevocably to *defect* if the other player ever elects *defect*. For further discussion of this and other iterations, see JEAN TIROLE, *THE THEORY OF INDUSTRIAL ORGANIZATION* §§ 6.3.1, 6.5.1 (1988).

⁷² To be perfectly rigorous, it could still be the case that platforms suffer from underadoption relative to a zero-transaction-cost world in which platform holders could write complete contracts that perfectly protect users against expropriation. At the very least, however, we do not live in a world that suffers from drastic or catastrophic underadoption.

legal action is costly and uncertain (and, given collective action constraints, likely to be severely underfunded whenever there is a diffuse user population); (iii) the transaction costs of entering into (and monitoring compliance with) contracts with large numbers of users may be prohibitive;⁷³ and (iv) specification costs may make it difficult to address all possible actions by which the host can expropriate value from users.⁷⁴ The last two points are especially relevant for a platform technology — in IT industry jargon, a “horizontal” application — that may be applied across a broad and difficult-to-anticipate range of users and uses. For example, it may be difficult to write, not to mention monitor, a contract that can specify all the ways in which a host could expropriate users’ investments through insufficient efforts to provide technical support, incomplete efforts at maintaining backward compatibility, or forward integration into users’ application markets.⁷⁵ Even if one implausibly assumes that all possible expropriation opportunities can be foreseen (or more precisely, can be foreseen at a reasonable cost), and that all contractual breaches can then be monitored and detected at a reasonable cost, it is still likely that definitional limitations would make it difficult to craft language that excludes expropriation opportunities without excluding other legitimate business actions or exposing the host to illegitimate claims by opportunistic litigants. The rapid pace of technological development, compounded by renegotiation difficulties, may further counsel hosts against taking on contractual obligations that are subsequently likely to be incompatible with changed conditions.⁷⁶ In sum, the gist of the matter is simply described: contract provides some meaningful ability by which to assure users, but contract cannot entirely, and may not even substantially, eliminate the host’s dilemma.

2. *Integration.* — Where contracts cannot be written to provide users with complete 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 miti-

⁷³ Note that transaction costs may be reduced through electronic contracting and monitoring technologies, which are already widely deployed in the ICT industry.

⁷⁴ On contractual incompleteness in the technology sector (in particular, the internet), see Shane Greenstein, *Glimmers and Signs of Innovative Health in the Commercial Internet*, 8 J. ON TELECOMM. & HIGH TECH. L. 25, 39–41 (2010); and in the software industry, see Schwarz & Takhteyev, *supra* note 14, at 613–14.

⁷⁵ The last covenant would raise concerns under antitrust law to the extent that a court could analogize any such contractual restraint to a horizontal agreement not to compete among actual or potential competitors. See, e.g., *Engine Specialties, Inc. v. Bombardier Ltd.*, 605 F.2d 1, 11 (1st Cir. 1979) (finding an agreement not to compete between a manufacturer and a distributor to be a horizontal agreement that was per se illegal, on the ground that the distributor was the manufacturer’s potential competitor).

⁷⁶ See Greenstein, *supra* note 74, at 40.

gated through vertical integration.⁷⁷ For example, the host can employ all developer users, which provides these users 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 strategy that technology leaders such as Microsoft, Oracle, and Cisco Systems, which regularly make acquisitions to procure "prepackaged" research and development, have implemented).⁷⁸

But integration suffers from (at least) four drawbacks. First, while the host can mitigate expropriation risk on the developer side, it cannot employ the other side of the market — end users, without whom most transaction paths cannot be completed. Second, an employment relationship cannot 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. This risk is particularly pronounced in light of the inherently uncertain range of possible applications for a platform technology, as compounded by the difficulty of anticipating end-user preferences in a retail market.⁷⁹ Multiple factors suggest that these 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 industry)⁸⁰ 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 "debug-

⁷⁷ On the ability of vertical integration to ameliorate opportunism risk, see WILLIAMSON, *supra* note 59, at 104.

⁷⁸ On serial acquisition strategies by these and other firms, see Sayan Chatterjee, *The Keys to Successful Acquisition Programmes*, 42 LONG RANGE PLAN. 137, 137–39 (2009).

⁷⁹ For examples of shared platform adaptation to changes in provider demand, see Eisenmann et al., *supra* note 15, at 155.

⁸⁰ The lower costs of integrating forward into all aspects of software development in the case of a vertical product anticipate (correctly) that forfeiture practices, including open source development, are observed far less often in those sectors of the industry. Hence, it may not be coincidental that OSS has achieved far more success in the case of horizontal (or "platform") software technologies, such as operating systems, and far less 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 uses, both contract specification and forward integration costs would be expected to be less onerous. For analogous observations (discussing the benefits of coalition-building for highly sophisticated end users who value customization), see Schwarz & Takhteyev, *supra* note 14, at 617.

ging," 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 development that enhances platform value.⁸¹ Like contract, integration is therefore an important but imperfect (and often extremely costly and even counterproductive) solution to the host's dilemma.

3. *Forfeiture*. — There exists an elegant but draconian solution to the commitment problem: the host can forfeit ownership 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 some access to technological knowledge at any price below its market value, including zero or negative prices. Hosts can undertake simple and complex forms of forfeiture actions, as discussed briefly below. The discussion below describes these forms at a high level of generality, which provides a framework for detailed description of the various forfeiture devices used in the OSS and smartphone markets, as presented in Part III.⁸²

(a) *Simple Forfeiture*. — The host may disclaim ownership over all or part of the platform technology. This form of forfeiture can be achieved through (i) disclosure of a technological asset that could otherwise be maintained as a secret and (ii) widespread, nonexclusive licensing of a technological asset at a nominal cost, when that asset could otherwise be licensed at market rates or to a more limited set of parties. Either action serves a simple but vital purpose: it gives users an asset that cannot be expropriated easily by the host, which in turn 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.⁸³ Poten-

⁸¹ See *supra* p. 1880.

⁸² In particular, see section III.B.1.a, pp. 1896–1906.

⁸³ Economists have identified rational incentives for monopolist sellers to invite competition. Professors Joseph Farrell and Nancy 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 T. Gallini, *Second-Sourcing as a Commitment: Monopoly Incentives to Attract Competition*, 103 Q.J. ECON. 673, 673–75 (1988). Other scholars have made related arguments. See, e.g., Economides, *supra* note 55, at 231 (arguing 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); Michael Kende, *Profitability Under an Open Versus a Closed System*, 7 J. ECON. & MGMT. STRATEGY 307, 320–21 (1998) (arguing that a systems firm can increase demand for the main component, and in some circumstances increase profits, by allowing competition in the aftermarket for sec-

tial rivals include users or any other party that 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 "dissident" noncompatible versions of the code that compete with or supplant the original version.⁸⁴ Even more severe consequences could result if a host gambles on a forfeiture strategy: it may lose control of the platform to rivals who never invested in its development. As IBM discovered in its inability to maintain exclusivity over the IBM PC (because third parties were able to reverse engineer its basic input/output system (BIOS) component without infringing the associated copyright⁸⁵), a firm that forfeits control over its core technology can lose the entire market to more adept producers of complementary goods within the consumption bundle.

(b) *Complex Forfeiture*. — In lieu of direct abandonment through disclosure or nonexclusive, nominal-cost licensing, the host can transfer control of the platform to a nonprofit organization, trade association, or some other neutral entity with which the host may retain some involvement, but over which it lacks unilateral control.⁸⁶ This strate-

ondary components, which constitutes a commitment to variety in the latter market); Andrea Shepard, *Licensing to Enhance Demand for New Technologies*, 18 RAND J. ECON. 360, 368 (1987) (arguing that "second sourcing," whereby a technology provider provides know-how to enable supply by a competitor, is a commitment to product quality). For an interesting practical application, see Gawer & Henderson, *supra* note 15, at 25-26, which uses evidence from Intel's history to illustrate that a monopolist platform holder has incentives to subsidize entry by third-party complementors in order to commit credibly against squeezing out other firms in the complements market.

⁸⁴ Forking actually led to the origin of the popular Firefox internet browser. As the Mozilla application suite was being developed by the open source community using Mozilla code (which had been developed on a proprietary basis by Netscape but then was 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 community leaders ultimately adopted as the primary supported version. See Jacques R. Bughin et al., *The Next Step in Open Innovation*, MCKINSEY Q., June 2008, at 113, 119.

⁸⁵ Following *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983), which clarified that operating system code is eligible for copyright protection, *id.* at 1253-54, other companies could not directly copy the BIOS code. However, they could attempt to reassemble it independently without infringing on the copyrights associated with the original BIOS code, a process which is known as "clean room" design. This practice was recently given some further latitude in *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000), which held that an emulator software developer's copying of a video game system manufacturer's code was protected under the fair use doctrine because the developer's reverse engineering process necessitated making intermediate copies of the manufacturer's code, *id.* at 603-04.

⁸⁶ For further discussion, see Stanley M. Besen & Joseph Farrell, *Choosing How to Compete: Strategies and Tactics in Standardization*, 8 J. ECON. PERSP. 117, 126 (1994). In another variant of this line of thought, Professors Annabelle Gawer and Rebecca Henderson observe that Intel attempts to commit against "squeezing" providers of complementary goods by establishing a non-profit unit (in corporate parlance, a "cost center") dedicated to generating and disseminating plat-

gy was adopted by AT&T as well as by other participants in the “Unix Wars” over setting the Unix standard in the 1980s: nonprofit organizations with publicly interested-sounding names (such as 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.⁸⁷ Since a nonprofit organization cannot distribute net earnings to any outside controlling interest,⁸⁸ and may be subject to the control of a diffuse membership, such an organization has reduced expropriation incentives and opportunities and, as a result, enjoys an increased ability to elicit users’ investments in the platform. Through the vehicle of a nonprofit or other non-investor-owned organizations (and, as discussed subsequently, by adopting publicly interested normative commitments), 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) cannot be excluded with sufficient certainty by contract. This rationale conforms to a broader proposition advanced by Professor 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.⁸⁹ At the same time, controlled forfeiture through a centralized organization can assure users that a single entity will have incentives and capacities to standardize and update the platform. In the absence of such a coordination mechanism, users may be discouraged from making investments in the platform given the prospect of technological instability or obsolescence that will endanger the value of those investments. In platform markets, the nonprofit structure and allied forms of centralized but noncontrolling organization provide a vehicle by which to commit against user expropriation and coordinate platform stability at a potentially lower cost than the alternatives of contract and integration.

form technologies among third-party complementors. See Gawer & Henderson, *supra* note 15, at 22–24.

⁸⁷ For further discussion, see section III.A.1, p. 1891.

⁸⁸ See BRUCE R. HOPKINS, *THE LAW OF TAX-EXEMPT ORGANIZATIONS* § 1.1(a), at 5, § 20.1, at 561 (9th ed. 2007). More 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 who has a close relationship with the organization” or “is in a position to exercise a significant degree of control over it.” *Id.* at 560; see also I.R.C. § 501(c)(3) (2006).

⁸⁹ See Henry B. Hansmann, *The Role of Nonprofit Enterprise*, 89 *YALE L.J.* 835, 844–45 (1980). These ideas are discussed at greater length in HENRY HANSMANN, *THE OWNERSHIP OF ENTERPRISE* 27–29 (1996).

III. ORGANIZATIONAL CONVERGENCE IN OPERATING SYSTEMS MARKETS

In selecting the preferred instruments by which to commit against opportunistic behavior, the host must compare the effectiveness of any instrument (or combination of instruments) in eliciting user adoption against the cost of adopting those instruments. Possible instruments include contract, integration, and forfeiture (in its various forms). Forfeiture is obviously the most potent means by which to commit against opportunism. As a stand-alone strategy, however, it is unworkable: the host cannot 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 nonforfeited 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 goods and services that are complementary to the platform. The host therefore faces a basic trade-off. 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, in order to accrue revenues to cover development and maintenance costs (and, in the case of a for-profit entity, in order to capture any remaining profits).

That trade-off yields a testable hypothesis. Host entities — both ostensibly for-profit and nonprofit entities — will select hybrid organizational forms that partially or completely forfeit control over the platform in order to maximize adoption gains while retaining partial control over the platform or complementary goods in order to minimize revenue losses. This Article now assesses that hypothesis against observed organizational strategies in operating systems markets, including a brief review of two paradigmatic “open” and “closed” models used historically and a detailed examination of two recently implemented hybrid models.⁹⁰ The imperative to commit against host opportunism, subject to the insolvency constraint, provides a general explanation for the controlled use of forfeiture strategies, and for the rough convergence of organizational forms used, by platform holders in these markets.

⁹⁰ For a more detailed and comprehensive overview of various models of software and hardware development, see generally West, *supra* note 15.

A. Old Models

1. *The Unix Model: Software as a Mostly Open Platform.* — The Unix operating system was developed in the early 1970s at Bell Labs.⁹¹ Its parent, AT&T, licensed Unix at nominal cost and distributed the source code to university researchers,⁹² who subsequently developed a number of variants.⁹³ In this model, the operating system acts as a “mostly open” platform that spawns third-party development of applications that enhance platform value. By the mid-1980s, Unix had become the industry platform in the minicomputer and workstation markets; however, its success was encumbered by the proliferation of incompatible versions.⁹⁴ 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.⁹⁵ This move was perceived by the market as an attempt to “reprivatize” Unix, so 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.⁹⁶ AT&T subsequently sold its interests in the Unix operating system to Novell, a commercial firm, which in turn transferred the rights to the UNIX trademark to the X/Open Consortium, a nonprofit industry consortium dedicated to standardizing Unix systems.⁹⁷ These repeated forfeiture actions promoted three common objectives: to place the operating system in the public domain, to induce investments by developers, and to enable the recovery of rents through the closed hardware components of the total consumption bundle. As described below, some of these hardware companies (notably IBM) are engaged

⁹¹ *Milestones in AT&T History*, AT&T, <http://www.corp.att.com/history/milestones.html> (last visited May 5, 2011).

⁹² See David McGowan, *Legal Implications of Open-Source Software*, 2001 U. ILL. L. REV. 241, 284.

⁹³ WEBER, *supra* note 20, at 39–43 (detailing the BSD variant created by the University of California, Berkeley).

⁹⁴ See SIMSON GARFINKEL ET AL., PRACTICAL UNIX AND INTERNET SECURITY 15–16 (2003); WEBER, *supra* note 20, at 95.

⁹⁵ See Garud & Kumaraswamy, *supra* note 25, at 355, 359–60.

⁹⁶ See GARFINKEL ET AL., *supra* note 94, at 16–17.

⁹⁷ *Id.* at 17. In 1996, X/Open merged with the Open Software Foundation to form the Open Group, a nonprofit association that now holds the UNIX trademark and maintains the set of standards for operating systems that qualify as Unix. *Id.* For completeness, note that the SCO Group, Inc. claims that it acquired the copyrights and other rights relating to the Unix source code from Novell, a position it has subsequently pursued in multiple infringement and related litigations against Novell and other entities involved in the Linux project. See *Frequently Asked Questions*, SCO GROUP, INC., <http://www.sco.com/scosource/linuxlicensefaq.html> (last visited May 5, 2011). The most recent relevant judgment (which is pending on appeal) found that the copyrights in dispute are owned by Novell. See Special Verdict Form at 1, SCO Grp., Inc. v. Novell, Inc., 721 F. Supp. 2d 1050 (D. Utah 2010) (No. 2:04-CV-139 TS), 2010 WL 2426012.

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.

2. *The Windows Model: Software as Semi-Closed Platform.* — The birth of the modern personal computer industry is often dated to a transaction between IBM and Microsoft, wherein Microsoft agreed to provide the MS-DOS operating system (crucially, on a nonexclusive basis) for IBM's new personal computer, launched in 1981. The remaining sequence of events 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.⁹⁸ After losing considerable market share to Windows, the successor to MS-DOS, IBM largely conceded defeat in the operating system market for desktop computing and ultimately exited the PC market entirely in a sale to Lenovo in 2005.⁹⁹ Microsoft retained exclusivity over the Windows platform, which persisted as a premium software product bundled with a premium microprocessor product sold by Intel and with 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,¹⁰⁰ who developed applications that enhanced the value of Windows relative to its competitors.¹⁰¹ 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.¹⁰² For Microsoft, prudent altruism has paid off handsomely: controlled forfeitures have enabled it to overcome the host's dilemma and realize the network ef-

⁹⁸ On this well-known history, see Gawer, *supra* note 58, at 62–63; and Henry W. Chesbrough & David J. Teece, *When Is Virtual Virtuous? Organizing for Innovation*, HARV. BUS. REV., Jan.–Feb. 1996, at 65, 68–70.

⁹⁹ See *Lenovo of China Completes Purchase of I.B.M.'s PC Unit*, *supra* note 47, at C5.

¹⁰⁰ See sources cited *supra* note 31; see also *United States v. Microsoft Corp.*, 84 F. Supp. 2d 9, 20 (D.D.C. 1999) (finding that outside developers had developed over 70,000 applications for Windows).

¹⁰¹ Industry commentators claim that millions of applications have been developed for the Windows system. See, e.g., Michael A. Cusumano, *Platforms and Services: Understanding the Resurgence of Apple*, COMM. ACM, Oct. 2010, at 23.

¹⁰² 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 L. Schmalensee, *Consumers Lose if Leading Firms Are Smashed for Competing*, in DID MICROSOFT HARM CONSUMERS? TWO OPPOSING VIEWS 97, 104 (David S. Evans et al. eds., 2000).

fects required to recoup its investment (and much more) in the platform and all extensions to it.

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 Microsoft adopted 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 the past decade, 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 (in part) by communities of volunteer programmers and (ii) the open source development of operating systems and other software applications by commercial entities. Two diametrically opposed motivations appear to drive these organizational models: altruistic volunteerism in the former case and commercial profit maximization in the latter. Closer inspection shows that these models roughly converge upon a common range of organizational forms: starting from different points of departure, both rely on some combination of corporate sponsorship to generate funding streams and nonprofit or other cooperative entities to manage and control core technological assets. Moreover, as shown subsequently, paid developers employed or otherwise funded by corporate sponsors have substantially eclipsed unpaid developers in their role on OSS projects. This organizational convergence is consistent with theoretical expectations: the underlying trade-off between platform forfeiture and control yields overlapping organizational structures that transcend 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 noninstrumentalist motivations behind uncompensated developer contributions, where reputational effects and intrinsic interest appear to play a strong motivating role and where ideo-

logical motivations play a weak motivating role.¹⁰³ The apparent rational choice puzzle posed by OSS is overstated in light of a few key developments. First, in the case of Linux, the leading open source project, “over 70% of all kernel development is demonstrably done by developers who are being paid for their work,”¹⁰⁴ who have been found to be the most productive contributors¹⁰⁵ and are particularly critical to the “core platform” components of open source projects.¹⁰⁶ Second, some of the most successful open source applications follow a

¹⁰³ See Rishab Aiyer Ghosh, *Understanding Free Software Developers: Findings from the FLOSS Study*, in PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE 23, 32–35 (Joseph Feller et al. eds., 2005); Karim R. Lakhani & Robert G. Wolf, *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects*, in PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE 3, 3 (Joseph Feller et al. eds., 2005) (noting that “enjoyment-based intrinsic motivation — namely, how creative a person feels when working [a] project — is the strongest and most pervasive driver” of OSS); Josh Lerner & Jean Tirole, *Some Simple Economics of Open Source*, 50 J. INDUS. ECON. 197, 212–20 (2002) (describing signaling and reputational incentives in the programmer market). It is worth noting that surveyed developers are likely to bias their *reported* incentives toward publicly interested motivations, especially given the “community ethos” of open source projects (something which the literature on OSS incentives tends to omit). In a suggestive analysis, David Lancashire marshals some support for the view that observed behavior by OSS developers is even more consistent with an instrumentalist model than survey data would appear to suggest. Per capita participation in the Linux and GNOME open source projects shows that open source projects tend to attract a disproportionate number of European programmers relative to U.S. programmers, which correlates inversely with the relationship between programmer salaries in those regions. See David Lancashire, *Code, Culture and Cash: The Fading Altruism of Open Source Development*, FIRST MONDAY, Oct. 3, 2005, <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/1488/1403>. The (potential) implication is that, consistent with instrumentalist behavior, the opportunity cost of foregone alternative activities influence programmers who participate in open source projects.

¹⁰⁴ GREG KROAH-HARTMAN ET AL., LINUX FOUND., LINUX KERNEL DEVELOPMENT 10 (2009), available at <http://www.linux.com/learn/whitepapers/doc/15/raw>; see also JONATHAN CORBET ET AL., LINUX FOUND., LINUX KERNEL DEVELOPMENT 13 (2010), available at http://www.linuxfoundation.org/docs/lf_linux_kernel_development_2010.pdf.

¹⁰⁵ See, e.g., 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, 995–96 (2006) (with respect to the Apache server software project); Evangelia Berdou, *Managing the Bazaar: Commercialization and Peripheral Participation in Mature, Community-Led Free/Open Source Software Projects* 139 fig.6-2, 154 fig.6-9 (June 2007) (unpublished Ph.D. dissertation, London School of Economics and Political Science) (on file with the Harvard Law School Library) (with respect to the GNOME and KDE user interface projects).

¹⁰⁶ See DAVE NEARY & VANESSA DAVID, NEARY CONSULTING, THE GNOME CENSUS: WHO WRITES GNOME? 21 & fig.4 (2010), available at http://www.neary-consulting.com/docs/GNOME_Census.pdf (reporting results of study on developer participation in the GNOME user interface project and finding that “paid contributors do the lion’s share of [work] in the core platform and middleware parts of the project, and unpaid developers tend to contribute much more in non-core applications . . . and developer tools,” *id.* at 21). These results substantially track findings in a previous case study of programmer contributions: in the case of the GNOME graphical user interface project, paid programmers were more likely to contribute to critical “core/platform” portions of the code base; in the case of the open source KDE user interface project, paid developers were more likely to maintain critical “core/platform” portions of the code base but no more likely to make contributions to those portions. See Berdou, *supra* note 105, at 139 fig.6-2 (on GNOME programmers); *id.* at 164 (on KDE project developers).

dual licensing model that distributes a free version but reserves technical support and proprietary features for paying customers.¹⁰⁷ Third (as discussed below), 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 a large mass of ideologically motivated volunteers¹⁰⁸ — both in the legal and (to a lesser but still surprising extent) economic literatures — do not accurately describe at least the most successful applications in the current market.¹⁰⁹ A publication in an IBM journal takes this view explicitly: “[T]he often quoted notion that such [open source] software is written primarily by people working *gratis* for the general good is false.”¹¹⁰ The substantial reprivatization of OSS development may disappoint its ideologically inspired proponents; however, it is an unsurprising outcome given the forfei-

¹⁰⁷ For an extensive discussion of licensing strategies (including several versions of dual licensing), see 451 GRP, OPEN SOURCE IS NOT A BUSINESS MODEL: HOW VENDORS GENERATE REVENUE FROM OPEN SOURCE SOFTWARE (2008). For further discussion of open licensing, see section III.B.1.a.i, pp. 1896–97.

¹⁰⁸ See sources cited *supra* note 17.

¹⁰⁹ Elsewhere I express similar doubts concerning the standard characterization of open source software. See Jonathan M. Barnett, *The Illusion of the Commons*, 26 BERKELEY TECH. L.J. (forthcoming 2011) (manuscript at 1806–15) (on file with the Harvard Law School Library). The data and analysis presented herein complement and extend that discussion. For other contributions expressing similar doubts and providing other evidence, see 451 GRP, *supra* note 107; and Stephen M. Maurer, *The Penguin and the Cartel: Rethinking Antitrust and Innovation Policy for the Age of Commercial Open Source* 1–9 (Univ. of Cal., Berkeley Goldman Sch. of Pub. Policy, Working Paper No. GSP10-006, 2010), available at <http://ssrn.com/abstract=1652292>. The extent to which, or the period during which, the standard characterization ever had any basis in fact remains unclear. As early as 2001, a trade commentator observed: “[T]he business of open source has finally come of age. Open-source software is in the IT marketplace alongside all the traditional . . . products.” Russell Pavlicek, *Trade Shows Grow Up*, INFOWORLD, Oct. 8, 2001, at 63, 63. At least, the notion that open source software is principally developed by a spontaneously organized mass of dispersed participants does not seem to have had much factual basis. A programmer survey released in 2000 found that open source code contributions rested on a narrow programmer base (ten percent of total authors wrote about seventy-two percent of code) and that Sun Microsystems was the second leading institutional contributor of code. See Rishab Aiyer Ghosh & Vipul Ved Prakash, *The Orbiten Free Software Survey*, FIRST MONDAY, July 3, 2000, available at <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/769/678>. Another survey released in 2002 found that, in the Apache open source project, a core group of approximately fifteen developers were responsible for “almost all new functionality” added to the code, while a somewhat larger group was responsible for generating fixes to reported defects. Audris Mockus et al., *Two Case Studies of Open Source Software Development: Apache and Mozilla*, 11 ACM TRANSACTIONS ON SOFTWARE ENGINEERING & METHODOLOGY 309, 322 (2002). Interestingly, that same study finds that the concentration of code contributions among developers in the Apache project was greater than the dispersion in selected commercial projects. See *id.* at 323. 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 to be entrusted with making code contributions.

¹¹⁰ P.G. Capek et al., *A History of IBM's Open-Source Involvement and Strategy*, 44 IBM SYS. J. 249, 257 n.4 (2005) (second alteration in original).

ture/control trade-off that precludes any perfect resolution to the host's dilemma.

(a) *Credible Commitment Through Controlled Forfeiture.* — A host entity can use the following forfeiture devices to commit credibly against future opportunism: (i) it can give away technological assets through contract, (ii) it can adopt social or ideological norms that covenant against user expropriation, and (iii) it can sequester technological assets in a foundation entity or other nonprofit or non-investor-owned form of organization. The combination of these devices constitutes an umbrella contract that governs the relationship between the host and user populations in any OSS project, resulting in some intermediate level of control/forfeiture with respect to the underlying platform and complementary set of goods and services. Contrary to conventional accounts in the legal literature that rely substantially on altruistic motivations,¹¹¹ these forfeiture actions can be accounted for by the fundamental imperative to commit against user expropriation, subject to the cost recovery constraint.

(i) *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 as a mechanism for committing against host expropriation. Making this commitment is especially vital in the case of an open source project, which, at least at its inception, has no resources with 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 comprised 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.¹¹² Moreover, the General

¹¹¹ For examples of this view, see sources cited *supra* note 17. To be clear, this Article does not deny the existence of altruistic and other noninstrumentalist motivations; it simply takes the view that (i) it is possible to account for the forfeiture practices that characterize OSS without reference to such motivations, and as discussed below in section III.B.1.a.ii, (ii) those (self-reported) motivations are best understood in strategic terms as commitment devices to address user lock-in.

¹¹² For similar observations, see Egon Franck & Carola Jungwirth, *Reconciling Rent-Seekers and Donators — The Governance Structure of Open Source*, 7 J. MGMT. & GOVERNANCE 401, 412–13 (2003), which contends that open source licensing strategies are “device[s] against fraud,” *id.* at 413; Merges, *supra* note 38, at 191–93, which argues that, through open source licensing, “open source contributors . . . restrict property claims of downstream contributors,” *id.* at 191; Joel West & Siobhán O'Mahony, *The Role of Participation Architecture in Growing Sponsored Open*

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 license,¹¹³ which ensures that (i) all developer users have access to all derivative applications distributed by other developer users and (ii) project management cannot exploit user contributions in order to develop proprietary products to which access will be constrained.¹¹⁴ The costs of altering the terms of the license further bolster these commitments. 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 under which all previous contributions were made would be prohibitively laborious.¹¹⁵ Put differently, the transaction costs of contractual

Source Communities, 15 *INDUSTRY & INNOVATION* 145 (2008), citing "increased public awareness," *id.* at 155, among other things, as benefits of open source licensing; and Siobhan Clare O'Mahony, *The Emergence of a New Commercial Actor: Community Managed Software Projects* (June 2002) (unpublished Ph.D. dissertation, Stanford University) (on file with the Harvard Law School Library), stating that "[t]he goal of [open source licensing] is to restrict unauthorized appropriation," *id.* at 26.

¹¹³ In 2007, the Free Software Foundation released revised version 3 of the GPL, which is less permissive in certain respects than version 2. *Compare GNU General Public License Version 3*, FREE SOFTWARE FOUND., INC. § 5(c) (June 29, 2007), <http://www.gnu.org/licenses/gpl.txt>, with *GNU General Public License Version 2*, FREE SOFTWARE FOUND., INC. § 2(b) (June 1991), <http://www.gnu.org/licenses/gpl-2.0.txt>. Reports are mixed over migration to GPL version 3. The influential Linux Foundation has declined to adopt version 3 due to the increased restrictions it would impose on commercial uses of Linux code. See Charles Babcock, *What Will Drive Open Source?*, *INFORMATIONWEEK*, Mar. 19, 2007, at 36. Furthermore, GPL version 3 currently constitutes only 6% of licenses used in open source projects (compared to 46% for GPL version 2), based on data updated daily in the Black Duck Software KnowledgeBase. See *Top 20 Most Commonly Used Licenses in Open Source Projects*, BLACK DUCK SOFTWARE, <http://www.blackducksoftware.com/oss/licenses> (last visited May 5, 2011). For the sake of simplicity, the discussion above refers to provisions in GPL version 2. See *GNU General Public License Version 2*, *supra*, § 2(b).

¹¹⁴ Other open source software is governed by more permissive licenses that place fewer or no constraints on the distribution of derivative applications. The leading permissive license is the Berkeley Software Distribution (BSD) license. See *The BSD License*, OPEN SOURCE INITIATIVE, <http://www.opensource.org/licenses/bsd-license.php> (last visited May 5, 2011). Important variants include the Apache license, which governs the popular Apache web server application, see *Licenses*, APACHE SOFTWARE FOUND., <http://www.apache.org/licenses> (last visited May 5, 2011), and the Mozilla Public License, which governs the popular Firefox browser application, see *Mozilla Licensing Policies*, MOZILLA FOUND., <http://www.mozilla.org/foundation/licensing.html> (last visited May 5, 2011). Permissive licenses have 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.

¹¹⁵ See O'Mahony, *supra* note 112, at 132 n.49.

amendment enhance the commitment signal constituted by the license's substantive content.

(ii) *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.¹¹⁶ 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 suggest “patches” to correct those defects.¹¹⁷ But a skeptical academic observer should consider whether the pronouncements of normative principle that accompany community contributions may be best understood as strategic tools with which to elicit a continuing flow of user contributions that are essential to project survival. In particular, ideologically formulated community norms may mitigate commitment concerns by enabling the host (and other users) to covenant credibly against expropriating users’ investments in the platform. 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. This rationale accounts for the fact that commercial firms that adopt open source strategies, and mature open source projects that rely heavily on corporate sponsorship, strive to develop and maintain a reputation for fairness and openness toward the developer community.¹¹⁸ Consistent with the host’s dilemma, any suggestion that an open source project will discriminate against community users — which occurred after Sun Microsystems’ acquisition of

¹¹⁶ See sources cited *supra* note 17.

¹¹⁷ See, e.g., Lenny T. Mendonca & Robert Sutton, *An Interview with Mitchell Baker*, MCKINSEY Q., Jan. 2008, at 12, 12–13.

¹¹⁸ See Adam G. Cohn & Gary Spiegel, *Effective Open Source Development Business Practices*, in OPEN SOURCE AND FREE SOFTWARE 2009, at 133, 137 (Lori E. Lesser et al. eds., 2009); Mario J. Madden, *Opening the Door: Four Questions to Ask in Developing an Open Source Software Policy*, in OPEN SOURCE SOFTWARE 2008, at 261, 274–75 (Stephen J. Davidson et al. eds., 2008); Mendonca & Sutton, *supra* note 117, at 3–4. If ideological fidelity to community norms is enforced by reputational mechanisms, then it may be argued that this is equivalent to the reputation-based mechanisms for “good behavior” that had previously been rejected as insufficient. See *supra* pp. 1883–84. The two can be distinguished, as follows: reputational mechanisms are a matter of simple cost-benefit analysis: in the platform technology setting, users assume the host is a repeat player who engages in cost-benefit calculation and identifies circumstances under which it nonetheless rationally defects in order to garner short-term expropriation gains; by contrast, morally flavored ideological mechanisms are not a matter of simple cost-benefit analysis and, precisely for that reason, allow the host to make a far stronger commitment to users against expropriative action. For an extensive discussion of moral principles as commitment mechanisms, see ROBERT H. FRANK, *PASSIONS WITHIN REASON: THE STRATEGIC ROLE OF THE EMOTIONS* 43–70 (1988).

the open source MySQL database in 2008 and recurred in anticipation of Oracle's acquisition of Sun in 2010¹¹⁹ — prompts emotional protest and endangers the continuing flow of user contributions. This phenomenon extends beyond the open source context. In general, dominant platform holders can overcome commitment difficulties by adopting morally formulated commitments that reassure third-party developers that the host will not expropriate their investments in the platform.¹²⁰ In short, ideology reduces to strategy.

(iii) *Foundation Entity*. — The organizational structure of open source projects has a key distinguishing element that gives them a competitive advantage over proprietary entities in committing against future opportunism. That advantage is the commitment power delivered by the legal safeguards and constraints embedded within the nonprofit organizational form. As shown in Table 2, nonprofit entities,¹²¹ which are subject to the control of advisory boards that are either self-appointing or elected by members,¹²² govern the development of leading open source applications. These leading open source applications include (i) the Linux operating system¹²³ (which, for 2010, constituted 17% of the worldwide operating system market for servers as measured by revenues, but a negligible percentage of the desktop computing market¹²⁴), (ii) the Ubuntu distribution (one of the leading non-

¹¹⁹ For a description of these transactions, see Ryan Paul, *Open Database Alliance Hedges Against Oracle Plans for MySQL*, ARS TECHNICA (May 14, 2009, 10:51 AM), <http://arstechnica.com/open-source/news/2009/05/open-database-alliance-hedges-against-oracle-plans-for-mysql.ars>.

¹²⁰ See Gawer & Henderson, *supra* note 15, at 26–27.

¹²¹ For other discussions of foundation entities in the open source context, see Siobhán O'Mahony, *Guarding the Commons: How Community Managed Software Projects Protect Their Work*, 32 RES. POL'Y 1179, 1190–93 (2003); West & O'Mahony, *supra* note 112, at 159–60; and O'Mahony, *supra* note 112, at 78–134, 167–193. While Professors Siobhán O'Mahony and Joel West tend to view the nonprofit entity as an instrument by which the volunteer programmer community protects its interests against corporate encroachment, this Article views the nonprofit entity as an instrument by which corporate sponsors commit to programmers and other users that they will not act contrary to the users' interests.

¹²² 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. See HANSMANN, *supra* note 89, at 17–18, 242.

¹²³ Strictly speaking, the name “Linux” applies only to 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. However, in general industry usage, “Linux” is used to refer both to the kernel and to the other software components required to form a complete operating system. 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 required to install and run Linux. See 2 THE INTERNET ENCYCLOPEDIA 488, 494–95 (Hossein Bidgoli ed., 2004).

¹²⁴ On Linux share in the server market, see *Worldwide Server Market Accelerates Sharply in Fourth Quarter as Demand for Heterogeneous Platforms Leads the Way*, According to IDC, BUSINESS WIRE (Mar. 1, 2011, 3:35 PM), <http://www.businesswire.com/news/home/20110228007267/en/Worldwide-Server-Market-Accelerates-Sharply-Fourth-Quarter> (releasing data on mar-

commercial Linux distributions¹²⁵), 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 nearly 22% of the worldwide browser market as of February 2011,¹²⁶ and the Apache server application, which constituted about 59% of the worldwide internet server market as of January 2011.¹²⁷

Other commentators have observed that a nonprofit entity is a useful logistical device for eliciting tax-deductible donations, providing a legal entity to hold intellectual property and other assets, and entering into contracts and other legal relationships.¹²⁸ 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, or the Internal Revenue Service (which can revoke tax-exempt status¹²⁹). That constraint in turn enables the

ket shares in the worldwide server operating system market, as measured by revenue as of the fourth quarter of 2010); and on Linux share in the desktop market, see Joe Brockmeier, *Linux Desktop Market Share: Small No Matter How You Measure*, NETWORK WORLD (Sept. 9, 2010, 2:59 PM), <http://www.networkworld.com/community/blog/linux-desktop-market-share-small-no-matter-ho> (reporting statistics on Linux share of desktop market in 2010, ranging from 1% to 4.9%). Note that if market share in the server operating system market is measured by number of units, then some sources provide substantially higher estimates of Linux market share. See *Operating System Software Used at Sites in All Locations January 2009*, NETCRAFT, https://ssl.netcraft.com/ssl-sample-report/CMatch/oscnt_all (last visited May 5, 2011) (estimating Linux market share, based on number of units, at 41%, based on survey of publicly accessible websites).

¹²⁵ Note that there are other important Linux distributions; Ubuntu has been selected 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 only ostensibly noncommercial, as it is largely funded as well as managed and operated by Canonical Ltd., which sells support services and even proprietary software products that are complementary to the Ubuntu distribution. See *Why Is It Free?*, UBUNTU, <http://www.ubuntu.com/how-can-it-be-free> (last visited May 5, 2011). For a full description of Canonical's services and products, see *About Canonical*, CANONICAL, <http://www.canonical.com/about-canonical> (last visited May 5, 2011).

¹²⁶ See Peter Bright, *Internet Explorer Share Surges, Firefox Wanes Based on New CIA Data*, ARS TECHNICA, <http://arstechnica.com/web/news/2011/03/internet-explorer-share-surges-firefox-wanes-with-new-cia-data.ars> (last visited May 5, 2011) (citing data current as of February 2011).

¹²⁷ See *January 2011 Web Server Survey*, NETCRAFT (Jan. 12, 2011), <http://news.netcraft.com/archives/2011/01/12/january-2011-web-server-survey-4.html>.

¹²⁸ See O'Mahony, *supra* note 121, at 1190–93; O'Mahony, *supra* note 112, at 81–82, 94.

¹²⁹ This possibility is not hypothetical. The IRS has initiated an audit with respect to certain royalty payments received by Mozilla. Those royalties, which represent approximately 90% of Mozilla's revenues, are paid by Google in exchange for 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, 5:00 PM), <http://www.pcworld.com/businesscenter/>

host to induce the user investment required to sustain platform adoption. Committing against opportunism 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. Developer users fear expropriation given the substantial funding received directly or indirectly from corporate sponsors, each of which (as indicated in Table 2) usually enjoys certain governance rights in the foundation.¹³⁰ 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.¹³¹ Reprivatization can be accomplished through various means: restricting participation in the “code sign-off” process,¹³² 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 through any of the foregoing methods. 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

article/154198/google_deal_produces_91_of_mozillas_revenue.html. For fiscal year 2008, those revenues (\$86.4 million) delivered a margin substantially in excess of Mozilla's expenses (\$49.4 million). See Mitchell Baker, *State of Mozilla and 2008 Financial Statements*, LIZARD WRANGLING — MITCHELL ON MOZILLA & MORE (Nov. 19, 2009), <http://blog.lizardwrangler.com/2009/11/19/state-of-mozilla-and-2008>.

¹³⁰ 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 request arose despite the fact that the community is currently “guided by an elected board of three Novell employees and two independent community contributors.” Ryan Paul, *OpenSUSE Linux Seeks Own Direction, More Autonomy from Novell*, ARS TECHNICA, <http://arstechnica.com/open-source/news/2010/06/opensuse-project-seeks-feedback-on-strategy-drafts.ars> (last visited May 5, 2011). If established, the openSUSE Foundation would constitute a nonprofit foundation nested within the Linux ecosystem, which is itself governed by the nonprofit Linux Foundation.

¹³¹ The acquisition prices paid for the most successful open source applications testifies to their commercial value: for example, on Sun's acquisition in January 2008 of the MySQL open source database for \$1 billion, see *Sun to Acquire MySQL*, MYSQL (Jan. 16, 2008), <http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html>; on Yahoo!'s acquisition in September 2007 of the open source Zimbra collaboration software product for \$350 million, see *Yahoo! Announces Agreement to Acquire Zimbra*, VMWARE ZIMBRA (Sept. 17, 2007), http://www.zimbra.com/about/zimbra_pr_2007-09-17.html; and on Novell's acquisition in January 2004 of Linux SUSE, a leading Linux distributor, for \$210 million, see Press Release, Novell, Inc., *Novell Completes Acquisition of SUSE Linux* (Jan. 13, 2004), <http://www.novell.com/news/press/archive/2004/01/pro4003.html>.

¹³² This process refers to the set of procedures and actions by which code contributions are approved for integration into the existing code. As this Article discusses subsequently, OSS projects usually operate under a strict hierarchy in which a small group of programmers participates in the sign-off process. See *infra* p. 1908. Moreover, the sign-off process in leading OSS projects is dominated by programmers employed by corporate sponsors. See *infra* Table 4, p. 1909.

TABLE 2: GOVERNANCE OF LEADING
OPEN SOURCE APPLICATIONS¹³³

<i>Product</i>	<i>Foundation (date est.)</i>	<i>License Type</i> ¹³⁴	<i>Governance Structure</i>	<i>Board Members (selected)</i> ¹³⁵
Apache server	Apache Software Foundation (1999)	Permis- sive	Membership enti- ty. Governing board elected by individual mem- bers, who are admitted by ma- jority vote of ex- isting members.	No formal corporate rep- resenta- tion. ¹³⁶

¹³³ All foundation entities are 501(c)(3) tax-exempt organizations under U.S. federal tax law, except for the Ubuntu Foundation, which is apparently organized as a trust entity based in the Isle of Man (as is its chief sponsor, Canonical). See BENJAMIN MAKO HILL ET AL., *THE OFFICIAL UBUNTU BOOK* 26, 29 (3d ed. 2008). 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.

¹³⁴ 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. Links to the licenses adopted by each foundation can be found in the Appendix.

¹³⁵ For purposes of this column, this Article lists selected entities that have a seat on the foundation board, the Advisory Board (in the case of the GNOME project), or the Community Council through an affiliated individual (in the case of the Ubuntu Foundation). All data on board membership are based on information found on each foundation's website as of March 7, 2011. See *About the GNOME Foundation*, GNOME FOUND., <http://foundation.gnome.org/about> (last visited May 5, 2011) (Gnome Foundation Advisory Board); *About the Mozilla Foundation*, MOZILLA FOUND., <http://www.mozilla.org/foundation/about.html> (last visited May 5, 2011) (Mozilla Foundation Board of Directors); *Board Members*, LINUX FOUND., <http://www.linuxfoundation.org/about/board-members> (last visited May 5, 2011) (Linux Foundation Board of Directors); *Board of Directors*, APACHE SOFTWARE FOUND., <http://www.apache.org/foundation/board> (last visited May 5, 2011) (Apache Software Foundation Board of Directors); *Members of "Ubuntu Community Council"*, UBUNTU CMTY. COUNCIL, <https://launchpad.net/~communitycouncil/+members> (last visited May 5, 2011) (Ubuntu Foundation Community Council). In the case of Ubuntu, the most notable individual affiliated with Canonical Ltd. is Mark Shuttleworth, Canonical's founder, owner, and former CEO.

¹³⁶ The Apache Foundation states that all board members are "individuals," *Frequently Asked Questions*, APACHE SOFTWARE FOUND., <http://www.apache.org/foundation/faq.html> (last visited May 5, 2011); however, as of 2005, IBM reported that two of the nine members are IBM employees, Capek et al., *supra* note 110, at 254.

<i>Product</i>	<i>Foundation (date est.)</i>	<i>License Type</i>	<i>Governance Structure</i>	<i>Board Members (selected)</i>
Firefox browser	Mozilla Foundation (2003)	Permissive	Self-appointing board.	No formal corporate representa- tion. ¹³⁷
GNOME user inter- face	GNOME Foundation (2000)	Weakly re- ciprocal	Membership entity. Govern- ing board elected by indi- vidual members. Advisory board includes sponsor representatives.	Canonical Ltd., Google, IBM, Intel, Motorola, Mozilla, Nokia, Novell, Oracle, Red Hat.
Linux kernel	Linux Foundation (2007)	Reciprocal	Membership entity. Repre- sentation rights on board tied to level of member- ship dues.	Fujitsu, Hitachi, IBM, Intel, NEC, Oracle, Qualcomm.

¹³⁷ As of fiscal years 2009 and 2008, Mozilla received, respectively, 86% and 91% of its royalty revenues (representing 71% and 80% of its total receivables, respectively) through a contract with Google, whereby Mozilla agreed to make Google the default search engine on its browser. See HOOD & STRONG LLP, MOZILLA FOUNDATION AND SUBSIDIARIES, DECEMBER 31, 2009 AND 2008: INDEPENDENT AUDITORS' REPORT AND CONSOLIDATED FINANCIAL STATEMENTS 16 (2010), available at <http://www.mozilla.org/foundation/documents/mf-2009-audited-financial-statement.pdf>. The 2008–2009 financial statements provide these revenue figures with respect to a “contract with a search engine provider” expiring in November 2011. See *id.* Based on previous press coverage, this almost certainly refers to the contract with Google. See Jason Kincaid, *Mozilla Extends Lucrative Deal with Google for 3 Years*, TECHCRUNCH (Aug. 28, 2008), <http://techcrunch.com/2008/08/28/mozilla-extends-lucrative-deal-with-google-for-3-years>. For further discussion, see *infra* note 129.

<i>Product</i>	<i>Foundation (date est.)</i>	<i>License Type</i>	<i>Governance Structure</i>	<i>Board Members (selected)</i>
Ubuntu (Linux distribution)	Ubuntu Foundation (2005)	Reciprocal	Members of Community Council and Technical Board nominated by Mark Shuttleworth (principal spon- sor), subject to approval by membership. ¹³⁸	Canonical Ltd. ¹³⁹

constrain the rights previously licensed under the GPL.¹⁴⁰ Or, as is often the case in hybrid licensing models (the most popular distribution model among commercial open source entities¹⁴¹), the entity could simultaneously license the code on an open source basis while selling a “user-friendly distribution” of the code, extensions to the code, complementary applications, or technical support and other services.¹⁴²

¹³⁸ See *About Ubuntu: Governance*, UBUNTU, <http://www.ubuntu.com/project/about-ubuntu/governance> (last visited May 5, 2011). Note that the Foundation is not actively engaged in the day-to-day operation and governance of the Ubuntu project, HILL ET AL., *supra* note 133, at 29; rather, by virtue of its announced \$10 million funding commitment, it provides assurance that the Ubuntu Linux distribution could be supported for some period of time independently of its current financial and operational dependence on its commercial sponsor, Canonical Ltd., see Dave Walker, *Ubuntu Unravalled*, LINUX USER & DEVELOPER (Mar. 31, 2010), <http://www.linuxuser.co.uk/opinion/ubuntu-unravalled>.

¹³⁹ On Ubuntu's dependence on Canonical (and its owner, Mark Shuttleworth) for funding and other support, see KEIR THOMAS ET AL., *BEGINNING UBUNTU LINUX: FROM NOVICE TO PROFESSIONAL* 22 (4th ed. 2009).

¹⁴⁰ See McGowan, *supra* note 92, at 300–02.

¹⁴¹ Based on a selected set of 114 open source–related commercial vendors, the authors of a recent study found that usage of hybrid licensing strategies breaks down as follows: (i) 14.9% of firms use a dual licensing strategy (that is, the same code base is licensed under free and proprietary versions), (ii) 23.7% of firms use an “open core” licensing strategy (that is, open source code is available on a free basis while proprietary extensions are available for a fee), and (iii) 14.9% of firms use an “open-closed” strategy (that is, open source products are distributed with positively priced complementary closed source products). Remaining firms in the sample released the code on a free basis only and generated revenue through the provision of support services (a purely open model) or release proprietary software that includes open source components (effectively, a closed model). See 451 GRP., *supra* note 107, at 11–13. That practice substantially eliminates any meaningful difference between open and proprietary software.

¹⁴² It might be unclear how a dual licensing strategy is consistent with the terms of an open source license. Note that mere provision of technical support poses no possible conflict, so any contractual conflict would arise concerning other product features. On that point, there are three

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 It's vitally important that Linux creator Linus Torvalds and other key kernel developers remain independent."¹⁴³ Not accidentally, Linus Torvalds, the project's founder, personally owns the Linux trademark,¹⁴⁴ which 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 the following: (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;¹⁴⁵ (ii) the Executive Director may not be an employee of any contributing member;¹⁴⁶ and (iii) amendment to the bylaws or dissolution of the foundation requires the vote of a majority of the directors.¹⁴⁷ Additionally, the Foundation's Advisory Board is not all powerful; rather, certain powers are delegated to a Technical Advisory Board, End-User Council, and Vendor Advisory Council. Each of these entities is mostly or exclusively populated by individuals who are affiliated with, or direct representatives of, corporate sponsors. But

possibilities in increasing order of complexity. 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 reciprocal and proprietary licenses or to license the "core" code base under a reciprocal license and provide an expanded code base under a proprietary license. The latter strategy requires that the copyright owner own or control all copyrights associated with the code (which can be achieved by requiring all contributors to enter into an assignment agreement, as is the case with respect to the open source Apache server application). See *Individual Contributor License Agreement V2.0*, APACHE SOFTWARE FOUND., <http://www.apache.org/licenses/icla.txt> (last visited May 5, 2011). Third, even in the absence of an assignment agreement, it may be possible to compartmentalize open source and closed source code in order to implement a dual licensing strategy that is at least arguably consistent with the terms of a reciprocal license such as the GPL. On this last (and more complex) possibility, see *infra* note 168.

¹⁴³ *About Us*, LINUX FOUND., <http://www.linuxfoundation.org/about> (last visited May 5, 2011). This statement should be taken with a grain of salt. The Foundation "sponsors" Torvalds as a "fellow"; the Foundation is in turn substantially governed by outside sponsors, which are therefore an indirect source of compensation. See *FAQ*, LINUX FOUND., <http://www.linuxfoundation.org/about/faq> (last visited May 5, 2011); *Staff*, LINUX FOUND., <http://www.linuxfoundation.org/about/staff> (last visited May 5, 2011).

¹⁴⁴ For the trademark registration at the U.S. Patent & Trademark Office, see *Trademark Electronic Search System (TESS)*, U.S. PATENT & TRADEMARK OFFICE, <http://tess2.uspto.gov/> (follow "Basic Word Mark Search (New User)" hyperlink; then search for "Linux"; then follow "Reg. Number 1916230" hyperlink).

¹⁴⁵ See Linux Found., Amended and Restated Bylaws of the Linux Foundation § 5.3(g) (Aug. 9, 2007) [hereinafter *Linux Bylaws*], available at <http://www.linuxfoundation.org/about/bylaws>.

¹⁴⁶ *Id.* § 6.7.

¹⁴⁷ See *id.* § 9.2.

each entity operates subject to a formal charter that specifies membership requirements, committee powers, and governance mechanisms that preserve various levels of influence for selected (although somewhat overlapping) constituencies in the Linux community.¹⁴⁸ This controlled diffusion of governance rights makes it difficult for any single sponsor (or multiple sponsors) to unilaterally direct foundation policy, to change foundation governance, or to dissolve the foundation entity. The transaction costs of charter amendment are uncharacteristically welcome: they bolster the commitment signal sent by the charter's substantive content.

(b) *Funding Controlled Forfeiture (or, 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 of supporting 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 voluntary contribution, (iii) cross-subsidy through revenue streams from complementary goods, and (iv) price discrimination across user populations. While the most successful open source applications appear to rely heavily on option (ii), they actually rely on option (iii) in a not-so-subtle disguise.¹⁴⁹ Table 3

¹⁴⁸ For further information, see *Advisory Councils*, LINUX FOUND., <http://www.linuxfoundation.org/programs/advisory-councils> (last visited May 5, 2011). For the composition of the Technical Advisory Board, see *Technical Advisory Board (TAB)*, LINUX FOUND., <https://www.linuxfoundation.org/programs/advisory-councils/tab> (last visited May 5, 2011). For the composition of the End User Council, see *End User Council*, LINUX FOUND., <http://www.linuxfoundation.org/programs/advisory-councils/euc> (last visited May 5, 2011). For the composition of the Vendor Advisory Council, see *Roster*, LINUX FOUND. (Jan. 19, 2009, 2:18 PM), <https://www.linuxfoundation.org/programs/advisory-councils/vac/roster>.

¹⁴⁹ For the sake of brevity, this Article omits 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, 1063 (2006); Geoffrey G. Parker & Marshall W. Van Alstyne, *Two-Sided Network Effects: A Theory of Information Product Design*, 51 MGMT. SCI. 1494, 1497 (2005).

TABLE 3: INSTITUTIONAL TOOLS FOR NON-EXCLUSIVE PLATFORM DESIGN

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

consolidates these funding mechanisms with the commitment mechanisms identified above. The table then sets forth the complete set of organizational elements that may be combined to implement the forfeiture/control trade-off in any institutional structure for developing and maintaining a platform good to which access is completely or substantially unconstrained. These elements can now be combined to generate the organizational design of the Linux operating system, which is set forth graphically below in a generic form that describes both the Linux project and other leading open source projects.

This complex picture is starkly different from the simple characterization of open source projects in much of the legal (and even some of the economic) literature as an altruistic and spontaneously organized mass of volunteers. Like other successful open source projects, Linux code development is governed by a strict hierarchy, in which a limited core of qualified developers (the “Core Developer Group” indicated above) develop code and approve changes to the code. These core developers are in turn assisted by reports of “bugs” and “fixes” contributed by a larger mass of participants.¹⁵⁰ Linux kernel development is overseen by the Linux Foundation and Torvalds, the project’s founder, who “remains the ultimate authority” on the incorporation of new code into the Linux kernel.¹⁵¹ The Foundation is supported by cash contributions, personnel, and other forms of support from 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 ten representatives appointed by corporate sponsors and four independent representatives.¹⁵² Sponsors’ contributions are consti-

¹⁵⁰ See WEBER, *supra* note 20, at 71; see also Andrea Bonaccorsi & Cristina Rossi, *Why Open Source Software Can Succeed*, 32 RES. POL’Y 1243, 1247 n.10 (2003) (on Apache and GNOME).

¹⁵¹ Staff, *supra* note 143.

¹⁵² Six large hardware firms are “Platinum” members, see *Members*, LINUX FOUND., <http://www.linuxfoundation.org/about/members> (last visited May 5, 2011), each of which makes a

tuted by membership dues — totaling \$5,405,000 in 2010,¹⁵³ admittedly a relatively paltry figure — but far more importantly, these contributions are also constituted by payments to firm personnel for buting code and participating in the “sign-off” process on admitting new code into the Linux kernel. Code contributions are important for two reasons: (i) they show a substantial monetary investment in the project (in the form of forfeited personnel hours) and (ii) they enable a firm to exert influence over the direction of the code (or selected portions thereof).¹⁵⁴ As shown in Table 4, paid contributors account for a disproportionate share of both submitted changes (almost 80%) and sign-offs (over 85%).¹⁵⁵ Note that the four leading sponsors are responsible for more than half of all sign-offs — a fact that is hardly consistent with the standard view of OSS projects as a spontaneously organized agglomeration of volunteer contributors. IBM in particular has made unilateral contributions to the Linux project (not to mention several other open source projects¹⁵⁶) that go far beyond its

\$500,000 annual contribution and is entitled to a seat on the board. Other firms at lower contribution levels collectively elect members to the board, *see* Linux Bylaws, *supra* note 145, § 5.3, sched. A. The current “Platinum” members are Fujitsu, Hitachi, IBM, Intel, NEC, Oracle, and Qualcomm. *See Members, supra*. The collectively elected board representatives are associated with AMD, NetApp, and Splashtop (formerly DeviceVM). *See Board Members, supra* note 135. For Foundation bylaws with respect to the appointment of board representatives, *see* Linux Bylaws, *supra* note 145, § 5.3(a), which describes rights of “Platinum” members to elect board directors individually, up to a maximum of ten directors; *id.* § 5.3(b), which describes rights of “Gold” members to elect up to three board members collectively; *id.* § 5.3(c), which describes rights of “Silver” members to elect one board member; and *id.* § 5.3(d), which describes rights of Technical Advisory Board to select one “at large” board member and rights of Individual Affiliates to elect two “at large” board members. Information on membership fees may be found in the Linux Bylaws.

¹⁵³ These calculations are based on sponsorship requirements as set forth in the Linux Bylaws, *see* Linux Bylaws, *supra* note 145, sched. A, and a list of corporate sponsors (with associated membership level) on the Linux Foundation website as of March 2011, *see Members, supra* note 152. Note that subscription fees are based on a tier schedule that increases as a function of the number of employees; in the case of a handful of member firms with an undetermined number of employees, the lowest subscription fee level was assumed.

¹⁵⁴ Interestingly, a recent study of code contributions to the GNOME graphical user interface project shows that firms “carve out” portions of the project by targeting their code contributions. That is, there is apparently a tacit division of labor among corporate contributors to the common platform. *See NEARY & DAVID, supra* note 106, at 19–20.

¹⁵⁵ These calculations are based on data available in KROAH-HARTMAN ET AL., *supra* note 104. Note that the Linux Foundation study assessed contributors’ affiliations based on the use of company email addresses, sponsorship information included in submitted code, and direct inquiries of contributors. As the study notes, this methodology may overstate the amount of corporate involvement given the possibility that programmers may do personal work through a company account; however, this inaccuracy is arguably counterbalanced by the inability to determine affiliation with respect to other contributors, who are then assumed to be contributing on a purely personal basis. *See id.* at 10.

¹⁵⁶ Perhaps most notably, in 2001, IBM launched the Eclipse software development tool at a reported cost of \$40 million, which it paid to acquire a startup that had developed the Eclipse

TABLE 4: CORPORATE CODE CONTRIBUTIONS
TO LINUX KERNEL DEVELOPMENT (JAN. 2008–JUNE 2009)¹⁵⁷

<i>Firm</i>	<i>Code Changes (percentage)</i>	<i>Sign-Offs (percentage)</i>	<i>Primary Market</i>
Red Hat	12	36.4	Services
IBM	6.3	5.3	Hardware; Services; Software
Novell	6.1	8.2	Services; Software
Intel	6.0	6.4	Semiconduc- tor chips
Oracle	3.1	1.2	Software
Fujitsu	1.5	—	Hardware
Google	0.8	10.5	Search services
Others	43.1	17.4	Various
TOTAL	78.9	85.4	

code. See Martin LaMonica, *Eclipse to Split from IBM*, CNET NEWS (Jan. 20, 2004, 7:39 AM), http://news.cnet.com/Eclipse-to-split-from-IBM/2100-7344_3-5143421.html. IBM then released the code under an open source license and, in 2004, it spun off the Eclipse group as an independent nonprofit entity, the Eclipse Foundation. See Taft, *supra* note 25. For a detailed analysis of the Eclipse project, see Maurer, *supra* note 109, at 4–9. 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. See David Berling, *Open Source: IBM's Deadly Weapon*, ZDNET (Apr. 8, 2002, 12:00 AM), <http://www.zdnet.com/news/open-source-ibms-deadly-weapon/296366>. As indicated previously, as of June 2010, IBM holds a board seat at the GNOME Foundation, see *supra* Table 2, pp. 1901–03, and IBM employees appear to be, or have been, members of the board of the Apache Software Foundation, see *supra* note 136.

¹⁵⁷ All information is based on data collected by the Linux Foundation. Figures shown reflect changes and sign-offs on Linux kernel versions 2.6.24 through 2.6.30, which were released during the period starting January 24, 2008, and ending June 9, 2009. See KROAH-HARTMAN ET AL., *supra* note 104, at 3, 11 & tbl.9, 13, 14 tbl.12. 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. In a December 2010 update to these findings, the rankings among the corporate contributors are mostly unchanged, although there are sometimes significant changes in absolute values. See CORBET ET AL., *supra* note 104, at 12–15.

membership dues and code contributions. In 2001, IBM pledged to provide \$1 billion in funding to Linux (which it claims to have recouped by 2002);¹⁵⁸ in 2005, it, with other corporate sponsors, founded and donated five hundred 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.¹⁵⁹ As of 2010, ten thousand IBM employees were working in Linux-related positions in R&D, sales, and marketing, including six hundred developers at the Linux Technology Center.¹⁶⁰ If one uses the figure of \$74,690 as the national mean salary for a programmer (as reported by the U.S. Bureau of Labor Statistics for 2009),¹⁶¹ then IBM's annual investment in the Linux Technology Center alone equals over \$44.8 million in salary expenses. Most recently, in June 2010, IBM and major semiconductor manufacturers founded a nonprofit foundation, the Linaro Foundation, 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.¹⁶² IBM's behavior may seem paradoxical: the world's leading patentee for the past ten 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. However, 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

¹⁵⁸ It has not been possible to independently verify the \$1 billion investment that IBM made in Linux 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. See Stephen Shankland, *IBM: Linux Investment Nearly Recouped*, CNET NEWS (Jan. 29, 2002, 9:00 PM), <http://news.cnet.com/2100-1001-825723.html>. Otherwise, IBM would have stopped making investments.

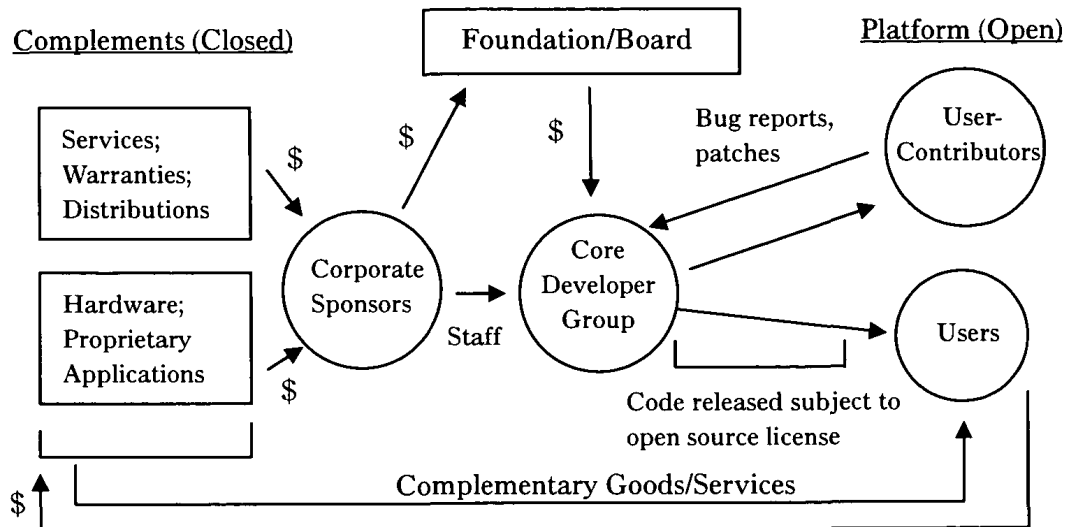
¹⁵⁹ See Martyn Williams, *IBM, Sony, Red Hat Join Others in Linux Patent Venture*, INFO WORLD (Nov. 10, 2005), <http://www.infoworld.com/t/platforms/ibm-sony-red-hat-join-others-in-linux-patent-venture-118>.

¹⁶⁰ See IBM, *WHY LINUX AND IBM: FLEXIBILITY, BUSINESS VALUE AND POWERFUL SOFTWARE* (2007), available at <ftp://ftp.software.ibm.com/linux/pdfs/WhyLinuxandIBMo507.pdf>.

¹⁶¹ *Computer Programmers*, BUREAU OF LABOR STATISTICS, U.S. DEP'T OF LABOR, OCCUPATIONAL EMPLOYMENT AND WAGES — MAY 2009, Part 15-1021 (Computer Programmers) (May 14, 2010), <http://www.bls.gov/oes/2009/may/oes151021.htm>.

¹⁶² See Press Release, Linaro, ARM, Freescale, IBM, Samsung, ST-Ericsson and Texas Instruments Form New Company to Speed the Rollout of Linux-Based Devices (June 3, 2010), <http://www.linaro.org/arm-freescale-ibm-samsung-st-ericsson-and-texas-instruments-form-new-company-to-speed-the-rollout-of-linux-based-devices>. The other founding members — ARM Holdings, Freescale Semiconductor, Samsung, ST-Ericsson, and Texas Instruments — are all semiconductor chip designers or manufacturers.

FIGURE 2: INSTITUTIONAL STRUCTURE
OF MATURE OPEN SYSTEMS



profit-seeking entities.¹⁶³ Viewed in the aggregate, the Linux project operates as a joint product development and marketing project that is fully funded and partially governed and 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 commoditized platform in the form of an operating system that can advance sales of (i) complementary hardware, (ii) other applications, or (iii) warranty, support, and consulting services to business end users.¹⁶⁴ This arrangement is fully consistent with business rationality: so long as revenues from proprietary goods exceed contributions to platform maintenance and support, sponsoring entities can anticipate a net positive return through participation in the development and maintenance of an open platform utility.

These complementary goods and services — some of which are set forth in Table 4 — generally fall into three categories: hardware, software, and services. First, firms such as IBM sell servers (a market in which IBM is the worldwide leader)¹⁶⁵ and other hardware that run

¹⁶³ For the leading statements of this view, see sources cited *supra* note 17.

¹⁶⁴ See Merges, *supra* note 38, at 192; Joel West & Scott Gallagher, *Challenges of Open Innovation: The Paradox of Firm Investment in Open Source Software*, 36 R&D MGMT. 319, 323–24 (2006).

¹⁶⁵ See Press Release, IBM, IBM Tops in Server Market in 4Q and Full Year 2009 (Feb. 24, 2010), <http://www-03.ibm.com/press/us/en/pressrelease/29517.wss> (reporting results released by IDC, an independent market analyst firm).

on the Linux system,¹⁶⁶ which is the chief competitor to Microsoft operating systems in the enterprise computing and server markets. To further advance its hardware sales, IBM adopts the typically generous policy of a host entity: it supplies technical support and even marketing assistance to third parties that develop applications compatible with IBM's Linux-based servers.¹⁶⁷ Other firms offer devices that have Linux code embedded in them: for example, a smartphone or a high-definition television may include Linux-based software to run various applications. Second, firms offer proprietary software extensions that are bundled with software based on the Linux code. For example, IBM and other firms sell software with embedded, open source code on a proprietary basis (such as the popular IBM WebSphere enterprise software suite, which bundles a proprietary application with the Apache open source web server application).¹⁶⁸ Third, firms such as Novell and Red Hat sell warranties, support services, and subscriptions to user-friendly Linux distributions such as SUSE

¹⁶⁶ See GABRIEL CONSULTING GRP., INC., *IBM & LINUX — 10 YEARS LATER* (2008), available at ftp://public.dhe.ibm.com/linux/pdfs/GCG_IBM_and_Linux-9_years_later.pdf; IBM, 2009 ANNUAL REPORT 23 (2010), available at http://www.ibm.com/annualreport/2009/2009_ibm_annual.pdf.

¹⁶⁷ See GABRIEL CONSULTING GRP., INC., *supra* note 166, at 7–8.

¹⁶⁸ See Ben Heskett, *IBM Fuels "Freeware" Efforts*, CNET NEWS (June 18, 1998, 6:50 PM) http://news.cnet.com/IBM-fuels-freeware-efforts/2100-1001_3-212482.html?tag=mncol;txt. This strategy is sometimes referred to in the trade literature as "open core" licensing; that is, the product combines open source code with closed source extensions (where the latter are reserved for paying users). See 451 GRP., *supra* note 107, at 12–13 (exploring related variants of this strategy). One may wonder how "Linux-embedded" hardware or software is compatible with the terms of an open source license. There are two answers. First, in the case of a "permissive" license such as the license that governs the popular Apache web server application and Firefox browser application, there is no obligation to distribute improvements on an "open source" basis. Second, in the case of a "reciprocal" license such as the GPL license (which governs Linux code), it may be possible to segregate the proprietary code from the open source code, such that only the latter is made available to users under the open source license. This second view is controversial in cases where the proprietary file links to a "GPL library," which could be captured by the GPL license under both an expansive understanding of the license and the protection afforded by copyright law to derivative works. The relevant section in GPL version 2 (the current predominant version) is decidedly ambiguous:

If identifiable sections of [a work based on the licensed code] are not derived from the Program [that is, the licensed code], and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

GNU General Public License Version 2, *supra* note 113, § 2. As an illustration of the unsettled nature of this question, the Software Freedom Conservancy, an advocacy organization, has sued several hardware manufacturers for violating the GPL by distributing devices with embedded Linux and failing to make (or offer to make) the source code available. See Beth Z. Shaw, *Recent Lawsuits Reflect Open Source Software Users' Copyright Compliance Obligations*, LEGAL BACKGROUNDER, May 7, 2010, http://www.wlf.org/publishing/publication_detail.asp?id=2164.

Linux Enterprise and Red Hat Enterprise Linux, respectively, while sponsoring “community” distributions that are available at no charge.¹⁶⁹ This strategy has been successful: contrary to conventional characterizations, a substantial percentage of Linux users pay for commercial distributions. Such payment is illustrated by Red Hat’s gross subscription revenue stream of nearly \$639 million in fiscal year 2009,¹⁷⁰ which nicely complements the other *unconventional* fact that Red Hat, a for-profit firm, is the leading contributor of code to the Linux kernel¹⁷¹ and the leading commercial source of “code commits” for the GNOME graphical user interface.¹⁷² For firms such as Red Hat, Novell, IBM, and others,¹⁷³ the open source model provides a collectively implemented mechanism by which to promote a commoditized platform technology, which induces outside development, which in turn enables the sale of complementary goods and services by those sponsor firms.

2. *The Nokia/Google Model: Software as Semi-Open Platform.* — Open models for software production start by addressing the credible commitment problem and must evolve to address the nonfunding problem. This proposition holds even for non-profit-seeking enterprises that are subject to an insolvency constraint: absent any funding so-

¹⁶⁹ Red Hat offers Red Hat Enterprise Linux at a positive price while sponsoring the community-supported Fedora project, which provides a free version. On the relationship between Red Hat and the Fedora project, see *Fedora Project Wiki*, FEDORAPROJECT, https://fedoraproject.org/wiki/Fedora_Project_Wiki (last visited May 5, 2011). For Red Hat’s Linux offerings, see RED HAT ENTERPRISE LINUX, <http://www.redhat.com/rhel> (last visited May 5, 2011). Novell offers SUSE Linux Enterprise at a positive price while sponsoring the community-supported openSUSE project, which provides a free version. On the relationship between Novell and openSUSE, see *OpenSUSE: Novell Involvement*, OPENSUSE, http://en.opensuse.org/openSUSE:Novell_involvement (last visited May 5, 2011). For Novell’s Linux offerings, see *SUSE Linux Enterprise*, NOVELL, <http://www.novell.com/linux> (last visited May 5, 2011).

¹⁷⁰ RED HAT, INC., 2010 ANNUAL REPORT 40 (2010), available at http://files.shareholder.com/downloads/RHAT/1049501295xox396285/ADF4C21A-48D1-4D91-8F43-8348C6D364FE/Red_Hat_2010_Annual_Report.pdf. Commercial users purchase subscriptions to Red Hat Linux (which uses source code that is otherwise freely available) not only for the associated support functions, but also because independent vendors of Linux-compatible software and hardware offer products that are “certified to” and supported with respect to a limited number of Linux distributions. For further discussion, see 451 GRP., *supra* note 107, at 21–22.

¹⁷¹ See *supra* Table 4, pp. 1909.

¹⁷² See NEARY & DAVID, *supra* note 106, at 16 tbl.2 (finding that Red Hat is responsible for 16.3% of total “code commits” over ten years of GNOME development).

¹⁷³ IBM explicitly describes its close relationship with the leading Linux distributors as follows: “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.” IBM, IBM IS COMMITTED TO LINUX AND OPEN SOURCE 3 (2008), available at ftp://ftp.software.ibm.com/linux/pdfs/IBM_and_Linux.pdf. 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 Press Release, Novell, Inc., Novell Finalizes IBM Investment (Mar. 23, 2004), <http://www.novell.com/news/press/archive/2004/03/pro4029.html>.

lution, the open model cannot cover platform development and maintenance costs. Closed models for software production start with a solution to the nonfunding problem and must evolve to address the credible commitment problem. Note that this principle is true even for profit-seeking enterprises that are subject to a more demanding profit-maximization constraint¹⁷⁴: absent a commitment device, the closed model can neither elicit nor 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 nonprofit enterprises to impose access restrictions notwithstanding ideological aspirations to the contrary.) Below, this Article shows how some for-profit competitors in the smartphone market — the site of the most recent battle to secure dominance for competing operating system platforms — have sought to secure market share by forfeiting core technology assets to nonprofit or other cooperative entities. At the same time, those same market pressures illustrate the solvency constraints that inherently limit the extent to which firms can give away those assets without securing a complementary revenue stream.

(a) *Nokia's Gifts (and Regrets)*. — Let us return to, and expand upon, Nokia's act of generosity with which this Article began. As shown in Table 5, Nokia invested nearly \$700 million in progressively acquiring full ownership of the Symbian operating system.¹⁷⁵ It then transferred management and distribution of the operating system to a nonprofit foundation governed jointly with its rivals in the handset manufacturing business and with telecommunications and semiconductor firms that compete with Nokia in the mobile telecommunications market¹⁷⁶ (see Table 6 for board members as of December

¹⁷⁴ 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.

¹⁷⁵ This figure is calculated as follows: in 1998, \$46 million to acquire a stake in the Psion operating system, simultaneously with Ericsson's and Motorola's purchases of equivalent stakes, see *Motorola Goes Ahead with Its 23% Stake in Symbian*, COMPUTERGRAM INT'L, Oct. 29, 1998, available at http://findarticles.com/p/articles/mi_moCGN/is_3527/ai_53149641; in 2003, \$65.7 million to acquire part of Motorola's interest, increasing Nokia's stake to 32.2%, see *Nokia and Psion Buy Motorola Out of Symbian*, COMPUTERGRAM INT'L, Oct. 9, 2003, available at <http://www.thefreelibrary.com/Nokia+and+Psion+Buy+Motorola+Out+of+Symbian.a0108661895>; in 2004, \$173.2 million to increase Nokia's stake to 47.9%, see Tony Cripps, *Symbian's Autonomy Assured as Owners Split Psion Stake*, COMPUTERGRAM INT'L, July 8, 2004, available at <http://www.thefreelibrary.com/Symbian's+Autonomy+Assured+as+Owners+Split+Psion+Stake.a0119040949>; and in 2008, \$410 million to buy out all remaining interests, see Hoover & McDougall, *supra* note 1, at 18.

¹⁷⁶ See Andrew R. Hickey, *Take that, Google Android: Nokia Creates the Symbian Foundation*, CRN (June 24, 2008, 10:30 AM), <http://www.crn.com/news/applications-os/index.htm> (view ar-

2010).¹⁷⁷ After spending two years to clear third-party rights, the foundation released the source code for the Symbian operating system to the public under an open source license, while Nokia reportedly continued to make the bulk of code contributions.¹⁷⁸ But, as shown below, even this exceptional forfeiture action omits considerable sums that Nokia invested directly or indirectly to acquire and then give away valuable technologies relating to the Symbian project. In 2006, Symbian (and hence, Nokia indirectly, at least partially) forfeited licensing revenues when it sold its user interface technology to Sony Ericsson — a direct rival of Nokia in the handset market — to reduce any perception of undue control by Nokia (then the largest stakeholder in Symbian).¹⁷⁹ In 2008, Nokia invested \$153 million to acquire Trolltech, a firm that held the rights to the open source “Qt toolkit,” a popular cross-platform software development tool that facilitates third-party development of applications for the Symbian operating system.¹⁸⁰ Toward this end, Nokia relicensed the toolkit under a more permissive license that enables third parties to use the toolkit to develop and distribute applications on a proprietary basis.¹⁸¹

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, and the resulting positive feedback effects on platform value, cannot 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, include handset manufacturers and telecommunications

chives for June 2008; then follow “Take That, Google Android: Nokia Creates The Symbian Foundation” hyperlink).

¹⁷⁷ As of December 17, 2010 (the date on which the Foundation was converted to a licensing entity), see *Symbian Foundation to Shut Down Websites*, LINUX PRO MAGAZINE (Nov. 29, 2010), <http://www.linuxpromagazine.com/Online/News/Symbian-Foundation-to-Shut-Down-Websites>, Nokia had only one representative on the Foundation’s board, which sat ten members in total, see Member Directory, Symbian Found., Board Members (on file with the Harvard Law School Library), despite having invested 100% of the capital required to acquire full ownership of the Symbian operating system, the foundation’s only asset. On the composition of the board at that time, see *infra* Table 6, p. 1925.

¹⁷⁸ See Jonathan Fildes, *Symbian Phone Operating System Goes Open Source*, BBC NEWS (Feb. 4, 2010, 12:02 AM), <http://news.bbc.co.uk/2/hi/technology/8496263.stm>.

¹⁷⁹ See Kevin Fitchard, *Symbian Sheds UIQ*, CONNECTED PLANET (Nov. 20, 2006, 12:00 PM), http://connectedplanetonline.com/mag/telecom_symbian_sheds_uiq.

¹⁸⁰ See Ryan Paul, *Nokia to Buy Trolltech, Will Become a Patron of KDE*, ARS TECHNICA (Jan. 28, 2008, 1:21 PM), <http://arstechnica.com/open-source/news/2008/01/nokia-buys-trolltech-will-become-a-patron-of-kde.ars>.

¹⁸¹ See Ryan Paul, *Nokia Qt LGPL Switch Huge Win for Cross-Platform Development*, ARS TECHNICA (Jan. 14, 2009, 1:00 AM), <http://arstechnica.com/open-source/news/2009/01/nokia-qt-lgpl-switch-huge-win-for-cross-platform-development.ars>.

TABLE 5: NOKIA'S GIFTS

<i>Date</i>	<i>Action</i>	<i>Cost</i>
June 1998	Nokia, Motorola, Ericsson, and Psion form Symbian to manage and develop Psion operating system	\$46M
Oct. 2003	Nokia purchases part of Motorola's interest in Symbian	\$65.7M
July 2004	Nokia increases ownership stake in Symbian to approximately 48%	\$173.2M
Nov. 2006	Symbian divests user interface technology to Sony Ericsson	undetermined
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; transfers management to Symbian Foundation	\$410M
Feb. 2010	After clearing third-party rights, Symbian Foundation releases Symbian source code under open source license	undetermined

operators with considerable bargaining power) that it has limited ability to expropriate user investments. Nokia sought to achieve this objective in two striking gambles. First, it adopted a nearly pure form of the open source model by transferring its core technological assets to an independent nonprofit foundation, which in turn disclosed the code to the outside developer community. Second, the foundation employed a cooperative architecture that diffused control over various platform features across multiple constituencies. The Board of Directors was positioned atop a federal structure consisting of a Feature and Roadmap Council, Architecture Council, User Interface Council, and Release Council. Each council operated subject to a formal charter instrument and was comprised of representatives from chip manufacturers, telecom operators, and handset manufacturers.¹⁸²

¹⁸² All of this information was obtained from the Symbian Foundation website, which included information on the representatives of each of the aforementioned bodies. As of December 17, 2010, the Symbian Foundation website is no longer in operation. Copies of all supporting materials concerning governance and other matters relating to the Foundation are on file with the Harvard Law School Library. As of December 15, 2010, AT&T, Fujitsu, Nokia, NTT DOCOMO, Qualcomm Innovation Center, Samsung, Sony Ericsson, ST Ericsson, Texas Instruments, and Vodafone constituted the Symbian Board; AT&T, NTT DOCOMO, Fujitsu, Nokia, Orange, QuIC, Samsung, Sony Ericsson, ST Ericsson, Telefonica, Texas Instruments, and Vodafone constituted the Features and Roadmap Council; Accenture, ARM, AT&T, China Mobile Communica-

Giveaway strategies to induce developer adoption are high-stakes gambles with no assured success. The risks were especially high in light of Nokia's aggressively open strategy to promote the Symbian platform. Netscape's near-complete loss of market share to Microsoft's Internet Explorer in the browser market illustrates that risk. Even dominant open systems may face competition from other open systems that can make equally credible commitments to users, while forfeiting valuable knowledge to competing closed systems that elect to "integrate around" the commitment problem. This dilemma roughly describes Nokia's predicament. In 2004, Nokia was the worldwide pioneer in the smartphone market, and its Symbian operating system represented about 65% of that market.¹⁸³ By 2010, Symbian was still the leading operating system in all new sales of smartphone handsets for the entire year — as shown in Figure 3 — but it had been overtaken by the Google-sponsored Android system in sales for the final quarter of 2010.¹⁸⁴ Android is sponsored by a Google-led coalition of firms known as the Open Handset Alliance and operates under a semi-open licensing and organizational arrangement.¹⁸⁵ Android's rapid

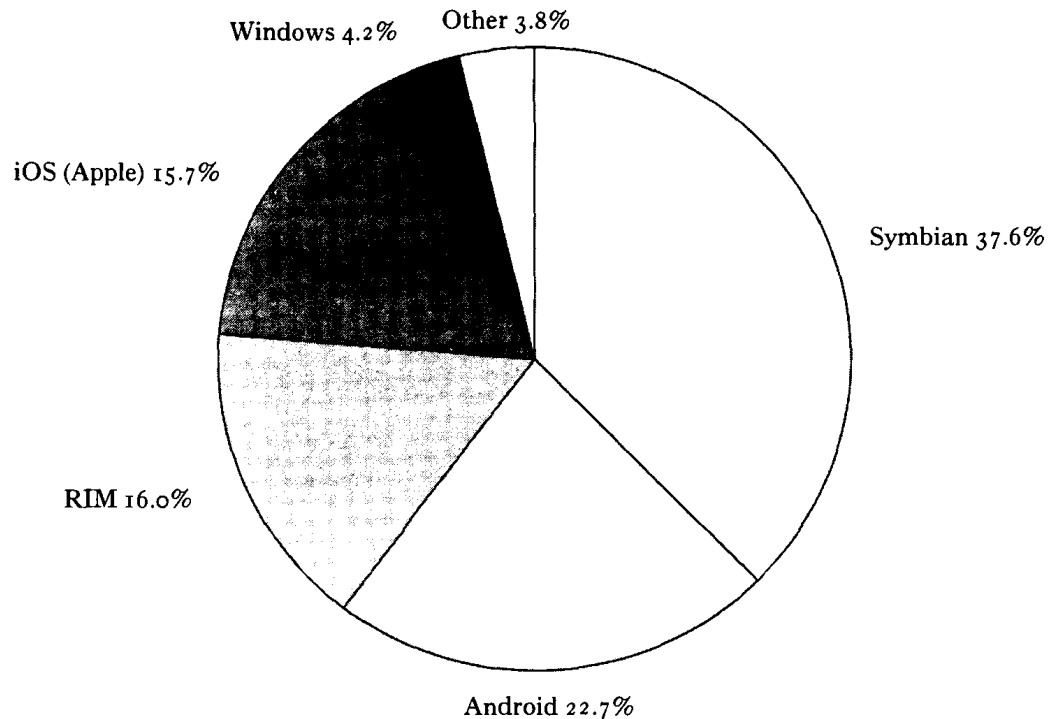
tions Corporation, DOCOMO, Fujitsu, Nokia, QuIC, Samsung, Sony Ericsson, ST Ericsson, and Texas Instruments constituted the Architecture Council; Adobe, AT&T, China Mobile Communications Corporation, NTT DOCOMO, Fujitsu, Nokia, Orange/France Telecom, QuIC, Samsung, Sasken Communication Technologies, Sharp, and Sony Ericsson constituted the User Interface Council; and Accenture, AT&T, Deutsche Telekom, NTT DOCOMO, Elektrobit, Fujitsu, Nokia, QuIC, Sony Ericsson, ST Ericsson, Teleca, and Texas Instruments constituted the Release Council.

¹⁸³ See Justin Hibbard, *Nokia vs. Microsoft in Mobile Phone Face Off*, RED HERRING (Mar. 4, 2004, 10:00 PM), <http://www.redherring.com/Home/5052>.

¹⁸⁴ See Press Release, Gartner, Inc., Gartner Says Worldwide Mobile Device Sales to End Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72 Percent in 2010 (Feb. 9, 2011), <http://www.gartner.com/it/page.jsp?id=1543014> (stating that Android had 22.7%, and Symbian 37.6%, market share of new sales of smartphone handsets for 2010, but that Android overtook Symbian sales for Q4 2010).

¹⁸⁵ For further discussion, see *infra* pp. 1921–22. Other Linux-based or Linux-compatible operating systems in the smartphone market include (i) the LiMo system, which is sponsored by the Linux Foundation and a number of telecommunications and handset providers and, as discussed subsequently, see *infra* note 197, includes both closed and open source layers; (ii) the MeeGo system, which was being developed jointly by Intel and Nokia on an open source basis but whose status is unclear since Nokia's adoption of the Windows Phone operating system, see Richard Adhikari, *MeeGo After Nokia: "I Will Survive" or "Where Did Our Love Go"?*, LINUXINSIDER (Feb. 15, 2011, 5:00 AM), <http://www.linuxinsider.com/rsstory/71862.html>; and (iii) Samsung's Bada platform, launched in November 2009, which offers a "kernel-configurable" architecture that can run on the Linux kernel or on a proprietary operating system. See *What Is Bada?*, SAMSUNG, <http://developer.bada.com/apis/docs/commonpage.do?menu=MC01140100> (last visited May 5, 2011); Press Release, Samsung Elecs. Co., Ltd., Samsung Launches Open Mobile Platform (Nov. 10, 2009), <http://www.bada.com/samsung-launches-open-mobile-platform>. Note that Palm's webOS system runs on the Linux kernel, see *Overview of HP WebOS*, HP WEBOS, <https://developer.palm.com/content/index.php?id=4292> (last visited May 5, 2011), and has released its source code, see *Open Source Packages*, PALM USA, available at <http://>

FIGURE 3: WORLDWIDE MARKET SHARE
FOR SMARTPHONE OPERATING SYSTEMS (2010)¹⁸⁶



climb in market share since its widespread release in 2009¹⁸⁷ has been driven by the familiar (but transient) source of platform dominance: widespread adoption by sophisticated intermediate users (including developers, carriers, and handset makers), which has in turn supported adoption by unsophisticated end users. For various reasons yet to be fully analyzed, Nokia has been unable to elicit comparable adoption of its more open Symbian system. Apparently aware of the historical volatility of dominant positions in platform markets, Nokia has recognized that its strong position in the worldwide handset market is now

opensource.palm.com/packages.html (last visited May 5, 2011). However, it is not entirely open. See Ken Hyers et al., *Palm Gives HP an Edge in Smartphones, Tablets*, WRAL TECH WIRE (May 9, 2010, 8:52 AM), http://localtechwire.com/business/local_tech_wire/news/blogpost/7569509. These other Linux-related systems have limited market presence.

¹⁸⁶ Figure 3 reflects market share breakdown based on operating systems used in new smartphone handsets sold worldwide for the year 2010. See Press Release, Gartner, Inc., *supra* note 184.

¹⁸⁷ See John Paczkowski, *Android Taking Smartphone Market Share from Everyone but Apple*, ALL THINGS DIGITAL (Nov. 3, 2010, 9:57 AM), <http://digitaldaily.allthingsd.com/20101103/android-taking-smartphone-market-share-from-everyone-but-apple>.

threatened.¹⁸⁸ Nokia's predicament is easily illustrated by a simple numerical comparison: as of February 2011, Nokia's online Ovi Store, a website offering applications for Nokia's handsets, had about 20,000 applications available, as compared to 314,644 applications available on Apple's App Store and over 160,000 applications available on Google's Android Market.¹⁸⁹ To press home the point, Nokia's newly appointed CEO, Stephen Elop, sent out an impassioned plea to the company's workforce in an internal memo: "We . . . are standing on a burning platform."¹⁹⁰

The looming risk of platform demise has prompted Nokia to take drastic preemptive action. On December 17, 2010, it converted the Symbian Foundation to an administrative entity responsible for licensing the Symbian trademark and related intellectual property while internalizing within Nokia the development and management of the Symbian operating system.¹⁹¹ Then, on February 11, 2011, Nokia took a dramatic step: it announced that it had adopted Microsoft's Windows Phone operating system as the primary platform for its smartphone handsets in lieu of Symbian.¹⁹² That is, Nokia had effectively exited its position as an independent provider in the smartphone operating system market. Following this Article's conceptual framework, competitive pressures apparently compelled Nokia first to constrain the openness of its Symbian platform asset, and second, to discard it altogether. These dramatic actions follow simple platform economics: if the host entity cannot induce collective development by giving away the platform, its only available strategies are either to exit the platform component of the relevant market or to bear the costs of developing the platform independently. Hoping to capitalize on the

¹⁸⁸ For an excellent and extensive history, see Georgina Prodhan & Tarmo Viki, *Welcome to Nokia, Mr. Elop*, REUTERS (Sept. 27, 2010, 6:25 AM), <http://www.reuters.com/article/idUSTRE68Q1IK20100927>.

¹⁸⁹ See DISTIMO, INSIGHTS INTO APPLE'S APP ECOSYSTEM: COMPARING MAC, IPAD AND IPHONE (2011); Prodhan & Viki, *supra* note 188; *Android Market Statistics*, ANDROLIB, <http://www.androlib.com/appstats.aspx> (last visited May 5, 2011).

¹⁹⁰ See *Full Text: Nokia CEO Stephen Elop's 'Burning Platform' Memo*, WSJ.COM (Feb. 9, 2011, 9:13 AM), <http://blogs.wsj.com/tech-europe/2011/02/09/full-text-nokia-ceo-stephen-elops-burning-platform-memo> (internal quotation marks omitted).

¹⁹¹ David Gilson, *Symbian Foundation to Close All Websites*, ALL ABOUT SYMBIAN (Nov. 29, 2010, 10:15 AM), http://www.allaboutsymbian.com/news/item/12332_Symbian_Foundation_to_close_al.php. On March 31, 2011, Nokia announced that it was making the latest version of the Symbian platform source code available to its "platform development partners." See Joao Luis, *Nokia's Symbian Website Now Open — Platform Source Code Made Available to Development Partners*, NOKIA-NEWS.COM (Mar. 31, 2011), <http://nokia-news.com/nokias-symbian-website-now-open-platform-source-codes-made-available-to-development-partners>.

¹⁹² See Kevin J. O'Brien, *Together, Nokia and Microsoft Renew a Push in Smartphones*, N.Y. TIMES (Feb. 11, 2011), <http://www.nytimes.com/2011/02/12/technology/12nokia.html>.

enormous resources of Microsoft, and its accumulated experience in cultivating a rich developer community in the desktop computing market, Nokia has largely elected the former option.

Nokia's internalization and then abandonment of Symbian are not inconsistent with the commitment concerns that often drive firms toward some combination of open and semi-open models in platform development and implementation. Assuming sufficient self-funding or external funding sources, closed or semi-closed strategies are viable organizational models that, at great expense, integrate forward in order to bypass the commitment problem (while avoiding the spillovers to rivals that are inherent to any forfeiture solution). Note, however, that even substantially closed models in the smartphone market make efforts to provide specifications and support to outside developers. This is true to varying degrees for every remaining leading provider in the smartphone operating system market: Google, which now represents the most open operating system, maintains the online Android Market;¹⁹³ Microsoft, consistent with its historical practice, releases Windows Mobile API specifications to developers and maintains the online "App Marketplace";¹⁹⁴ and Apple, in a deviation from its historical practice, (as noted above) maintains the online "App Store," where developers can post iPhone applications (subject to approval by, and revenue sharing with, Apple).¹⁹⁵ Even RIM, which arguably operates the most closed system with respect to its BlackBerry device, has belatedly undertaken a similar initiative in order to induce outside development.¹⁹⁶ These hybrid permutations conform to theoretical expectations: market pressures both push closed systems to incorporate some degree of openness in order to commit against user expropriation and, as the history of the Nokia platform illustrates, drive open systems to impose access restraints when required to preserve a positive funding stream.

(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).

¹⁹³ See *Android Market*, GOOGLE, <https://market.android.com> (last visited May 5, 2011).

¹⁹⁴ See *Windows Phone 7 Apps*, MICROSOFT, <http://www.microsoft.com/windowsphone/en-us/apps/default.aspx> (last visited May 5, 2011); see also MICROSOFT, HOW TO WRITE MANAGED CODE THAT USES THE MOBILE BROADBAND API 3 (2009), available at http://www.microsoft.com/whdc/connect/wireless/MB_ManagedCode.msp.

¹⁹⁵ See Kevin J. Boudreau & Karim R. Lakhani, *How to Manage Outside Innovation*, MIT SLOAN MGMT. REV., Summer 2009, at 68, 75.

¹⁹⁶ See Phred Dvorak, *RIM Tries Harder on Apps*, WSJ.COM (Oct. 15, 2010, 9:07 AM), <http://online.wsj.com/article/SB10001424052748703631704575552262869819460.html>.

While an implicit consortium supports the development of Linux for the enterprise computing market, explicit consortia support (or have supported) the development of open source operating systems for the smartphone market. Even following the closure of the Symbian Foundation, two consortia or similar multifirm arrangements are currently in operation in this market. First, twenty-seven handset makers and telecommunications service providers formed the LiMo ("Linux Mobile") Foundation in 2007 in order to establish a nonproprietary Linux-based operating system for the smartphone market.¹⁹⁷ The LiMo Foundation, whose members collectively represent hundreds of millions of mobile telephone subscribers¹⁹⁸ and whose board of directors includes representatives from leading handset manufacturers and telecommunications providers,¹⁹⁹ operates subject to detailed bylaws. Like the Linux Foundation and the Symbian Foundation, it disperses decisionmaking power over platform development across multiple committees, control over which is in turn allocated among groups of handset makers and telecommunications operators.²⁰⁰ Second (and with considerably greater market success to date), the eighty-member²⁰¹ Open Handset Alliance (OHA) was formed in 2007 as a loosely organized association of handset makers, telecommunications service providers, and other technology firms with the express goal of promoting the nonproprietary, Linux-based Android operating system for the smartphone market.²⁰² The OHA's governance and develop-

¹⁹⁷ See *History*, LIMO FOUND., <http://www.limofoundation.org/en/history.html> (last visited May 5, 2011). For current membership, see *Current Members*, LIMO FOUND., http://www.limofoundation.org/component/option,com_limomembers/Itemid,134 (last visited May 5, 2011). "Nonproprietary" is used rather than "open source" because the LiMo Foundation restricts access to the source code behind its platform to firms that 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 Found., *Bylaws of LiMo Foundation Annex A*, art. VI (2009) [hereinafter *LiMo Bylaws*], available at http://www.limofoundation.org/images/stories/pdf/090928_pub_limo_bylaws_consolidated_as_of_september_28th_09.pdf. 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 application protocol interfaces available to third-party developers on a royalty-free basis, see *id.* Annex A, art. III.B.2.

¹⁹⁸ See W. David Gardner, *LiMo Foundation Seeks Alliance with WAC*, INFORMATIONWEEK (Mar. 2, 2010, 4:07 PM), http://www.informationweek.com/news/software/open_source/showArticle.jhtml?articleID=223101231.

¹⁹⁹ The board of directors includes representatives from NEC, NTT DOCOMO, Panasonic, Samsung, SK Telecom, Telefonica, and Vodafone. For a full list, see *Governance*, LIMO FOUND., <http://www.limofoundation.org/en/governance.html> (last visited May 5, 2011).

²⁰⁰ See *LiMo Bylaws*, *supra* note 197, art. 3 (on the board of directors); *id.* art. 7 (on the management councils).

²⁰¹ This figure reflects membership as of May 5, 2011. See *Members*, OPEN HANDSET ALLIANCE, http://www.openhandsetalliance.com/oha_members.html (last visited May 5, 2011).

²⁰² See Marguerite Reardon, *Google Unveils Cell Phone Software and Alliance*, CNET NEWS (Nov. 5, 2007, 9:21 AM), http://news.cnet.com/8301-17939_109-9810937-2.html?tag=mncl;txt.

ment structure is more closed than are those of the LiMo Foundation, the Linux Foundation, or the now-defunct Symbian Foundation. Google reportedly controls development of the Android code (purchased by Google from a third party in 2005²⁰³), 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 handsets.²⁰⁴ Moreover, by agreement with the relevant carriers, those handsets often include proprietary services or other applications offered by Google (for example, Google's popular email application Gmail).²⁰⁵ Despite Google's in-house code development and incorporation of proprietary applications, the Android platform does offer a meaningfully open environment insofar as it provides developers with the ability to download the source code together with a software development kit at no charge, which has facilitated the development of thousands of applications available online on the Android Market. Following standard platform economics, the investments made by sophisticated developer users in turn propel adoption by unsophisticated end users.

If one consolidates membership in these explicit consortia (the LiMo Foundation, the Open Handset Alliance, the previously mentioned Linaro Foundation, and for historical completeness, the now non-operational Symbian Foundation) with the implicit consortium constituted by the Linux Foundation, one can draw a more complete picture of the function that nonprofit and other cooperative forms of organization play in the competition for platform dominance in the smartphone

For a list of members, see *Members*, *supra* note 201. The OHA's governance structure is difficult to confirm due to the absence of formal constituting documents or publicly available contractual instruments. 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 David Meyer, *Google 'Guarantees' Android Compatibility*, ZDNET UK (Nov. 13, 2007, 8:02 AM), <http://news.zdnet.co.uk/communications/0,1000000085,39290713,00.htm>.

²⁰³ See Stefanie Olson, *Google Buys Android*, CNET NEWS (Aug. 17, 2005, 1:35 PM) http://news.cnet.com/8301-10784_3-5837102-7.html.

²⁰⁴ 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." *Frequently Asked Questions*, ANDROID OPEN SOURCE, <http://source.android.com/faqs.html> (last visited May 5, 2011). The website also provides a lengthy answer to the question, "Why is Google in charge of Android?" See *id.* Note that Google programmers appear to develop almost all of the Android code, as indicated by a review of the list of leading contributors to the project. See LiMo Found., *Mobile Open Source Economic Analysis* 21 (Aug. 2009), available at <http://www.limofoundation.org/images/stories/pdf/limo%20economic%20analysis.pdf>. 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), <http://www.crunchgear.com/2009/11/14/its-googles-world-and-handset-makers-just-live-in-it>.

²⁰⁵ See Matt Richtel, *Google: Expect 18 Android Phones by Year's End*, N.Y. TIMES.COM (May 27, 2009, 7:20 PM), <http://bits.blogs.nytimes.com/2009/05/27/google-expect-18-android-phones-by-years-end>.

market. Remarkably, vital intellectual assets in the lucrative smart-phone market are held by nonprofit or other consortia, which then sometimes release those assets with few contractual restrictions to the broader developer community under an open source license. Table 6 lists firms that hold board positions or that are otherwise “material” participants in these consortia.²⁰⁶

It is worth noting the types of firms that have elected to participate in these consortia: principally, handset makers, telecommunications providers, and semiconductor chip providers. These firms 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 backward into the operating system market. Each participant relinquishes its ability to secure revenues by regulating access to the platform, which is reduced to a privately operated utility 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

²⁰⁶ With respect to each foundation, the table lists a firm only if it is a board member in any single foundation or a “material participant” in at least two foundations (in each case, based on information available on each foundation’s website either as of March 3, 2011, or in the case of the Symbian Foundation, as of December 15, 2010). Despite its recent conversion to a licensing entity, the Symbian Foundation is included for completeness. Material participants are designated by an “X.” “(B)” indicates that a firm has at least one representative on the board of a foundation. A firm may qualify as a “material participant” in several ways. For the LiMo Foundation, only board members are listed. For the Linaro Foundation, a material participant is a firm that is listed as a Commercial Sponsor. For the Linux Foundation, a material participant is a firm that is listed as a Platinum or Gold member (determined by membership dues). The OHA makes no membership distinctions and does not appear to require any membership fees; hence, all members are considered “material participants” for purposes of this table. For the Symbian Foundation, a material participant is a firm that was listed as a Founder or Core member (determined by membership dues). Although China Mobile, HTC, Sprint, and T-Mobile are not material participants in any consortium other than the OHA, they have been included in this table because they appear to have played prominent roles in the initial manufacture and distribution of Android-enabled phones. These firms have been marked with an “X*.” See Biggs, *supra* note 204 (regarding HTC); Tim Conneally, *China Mobile Launches ‘OPhone’ to Counter China Unicom’s iPhone*, BETANEWS (Aug. 31, 2009, 3:11 PM), <http://www.betanews.com/article/China-Mobile-launches-OPhone-to-counter-China-Unicom-iPhone/1251745874> (regarding China Mobile); Nancy Gohring, *Sprint to Sell Android Phone in October*, PCWORLD (Sept. 3, 2009, 1:50 PM), http://www.pcwORLD.com/article/171402/sprint_to_sell_android_phone_in_october.html (regarding Sprint, T-Mobile, and HTC). With respect to HTC in particular, see Bruce Einhorn et al., *A Former No-Name from Taiwan Builds a Global Brand*, BLOOMBERG BUSINESSWEEK, Nov. 1–7, 2010, at 37. For reasons of space, this table omits the following firms: ACCESS (a LiMo board member), SK Telecom (a LiMo board member), NetApp (a Linux board member), Splashtop (a Linux board member), and ST Microelectronics (a Symbian board member). A full list of corporate membership in the boards of directors across all entities is available from the author upon request.

TABLE 6: MATERIAL CORPORATE PARTICIPANTS
IN NONPROFIT AND OTHER CONSORTIA RELATING
TO THE SMARTPHONE MARKET

		Foundation (date est.)				
		<i>Linux</i> (2007)	<i>Symbian</i> (2007)	<i>LiMo</i> (2007)	<i>OHA</i> (2007)	<i>Linaro</i> (2010)
Handset Vendors	<i>Fujitsu</i>	X(B)	X(B)			
	<i>Hitachi</i>	X(B)				
	<i>HTC</i>				X*	
	<i>Motorola</i>	X(B)			X	
	<i>NEC</i>	X(B)		X(B)	X	
	<i>Nokia</i>	X	X(B)			
	<i>Panasonic</i>			X(B)		
	<i>Samsung</i>		X(B)	X(B)	X	X
	<i>Sony</i> <i>Ericsson</i>		X(B)		X	
Telecoms	<i>AT&T</i>		X(B)			
	<i>China</i> <i>Mobile</i>				X*	
	<i>NTT</i> <i>DOCOMO</i>		X(B)	X(B)	X	
	<i>Sprint</i>				X*	
	<i>Telefonica</i>			X(B)	X	
	<i>Vodafone</i>		X(B)	X(B)	X	
	<i>T-Mobile</i>				X*	
Chip Vendors	<i>AMD</i>	X(B)				
	<i>ARM</i>				X	X(B)
	<i>Freescale</i>				X	X
	<i>Intel</i>	X(B)			X	
	<i>Qualcomm</i>	X(B)	X(B)		X	
	<i>ST-Ericsson</i>				X	X
	<i>Texas</i> <i>Instruments</i>		X(B)		X	X(B)
Other	<i>Google</i>	X			X	
	<i>IBM</i>	X(B)				X(B)
	<i>Oracle</i>	X(B)				

systems (Microsoft, RIM, Apple).²⁰⁷ 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 and 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 of other participants and of the broader population of developer users who must be induced to invest in the platform in the first place. The absence of any controlling interest accelerates intermediate users' investments 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 in order to create a common platform that enables its members to compete over the remaining differentiated portions of a smartphone device.²⁰⁸ Large-scale forfeiture of the most valuable technology assets in one of the world's most valuable markets is fully consistent with 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? The following Part now turns to that question.

IV. IMPLICATIONS: WHAT'S SO GOOD ABOUT "FREE"?

Two propositions are routinely asserted or implied in legal, economic, and policy (and even some business) commentary on OSS development 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. These linked propositions appear to drive policy proposals, and in some cases government actions, to promote OSS adoption through subsidies for OSS or other forms of open technological development, to promote procurement preferences for OSS and related innovation models over proprietary alternatives, and to encourage other means by which to influence market outcomes.²⁰⁹ A number of intellectual property scholars have

²⁰⁷ As described previously, even these systems are not entirely closed; rather, they incorporate substantial open elements. See *supra* p. 1920.

²⁰⁸ See LiMo Found., Intellectual Property Policy (July 7, 2009), http://www.limofoundation.org/images/stories/pdf/090707_limo_ipr_policy_final.pdf.

²⁰⁹ For descriptions (but not endorsements) of these proposals, see GOVERNMENT POLICY TOWARD OPEN SOURCE SOFTWARE (Robert W. Hahn ed., 2002); Jyh-An Lee, *New Perspectives on Public Goods Production: Policy Implications of Open Source Software*, 9 VAND. J. ENT. & TECH. L. 45 (2006); Klaus M. Schmidt & Monika Schnitzer, *Policy Subsidies for Open Source? Some Economic Policy Issues of the Software Market*, 16 HARV. J.L. & TECH. 473 (2003); Sebas-

advocated government intervention to promote OSS development or other (ostensibly) open innovation models in order to constrain control by private enterprise over the free exchange of technological knowledge.²¹⁰ This line of argument tends to rely on a simple underlying assumption — namely, privately interested for-profit entities seek to exert maximal control over technological assets in order to earn monopoly rents while publicly interested nonprofit entities lift all controls over technological assets in order to maximize access.²¹¹ It therefore easily follows that any government intervention in order to promote market adoption of open over closed innovation structures must improve social welfare by opening up access, so long as there is no sufficiently adverse effect on innovation incentives. Given that the mass voluntary participation elicited by OSS projects “proves” that the final qualifying condition is not a cause for concern under certain typical circumstances, the case for open innovation models is virtually sealed.²¹²

Perhaps that line of argument in favor of open innovation models is true in theory under certain assumed conditions.²¹³ But, with respect to the leading OSS projects that proponents of this line of argument cite as primary real-world illustrations (*and* with respect to many of the leading closed models that those proponents cite as primary real-world *counter*-illustrations), it is *not* true in practice. The standard association of closed models with privately interested for-profit entities and open models with publicly interested nonprofit entities does not track the mixed organizational strategies that prevail in platform technology markets. Evidence concerning the funding, staffing, and governance of operating systems development in the enterprise computing and smartphone markets, as well as consistent evidence of access giveaways throughout the historical development of platform technology markets, is compelling. Ostensibly profit-motivated and non-profit-motivated participants tend to converge on a common range of organi-

tian von Engelhardt & Stephen M. Maurer, *The New (Commercial) Open Source: Does It Really Improve Social Welfare?* 6–7 (Univ. of Cal., Berkeley Goldman Sch. of Pub. Policy, Working Paper No. GSPP10-001, 2010), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1542180; and Maurer, *supra* note 109, at 35.

²¹⁰ See, e.g., Yochai Benkler, Lecture, *Freedom in the Commons: Towards a Political Economy of Information*, 52 DUKE L.J. 1245, 1275 (2003); Lawrence Lessig, *Open Source Baselines: Compared to What?*, in GOVERNMENT POLICY TOWARD OPEN SOURCE SOFTWARE 50, 54–67 (Robert W. Hahn ed., 2002); Eben Moglen, *Free Software Matters: Free Government, II*, EBEN MOGLEN (Oct. 30, 2002), <http://emoglen.law.columbia.edu/publications/lu-24.html>.

²¹¹ See, for example, LAWRENCE LESSIG, *THE FUTURE OF IDEAS: THE FATE OF THE COMMONS IN A CONNECTED WORLD* (2001); and Zittrain, *supra* note 12, who rely throughout on consistent associations of closed models with the private interest and open models with the public interest.

²¹² For leading proponents of this view, see sources cited *supra* note 17.

²¹³ See sources cited *supra* note 17.

zational structures that mix open and closed elements in various permutations to achieve an efficient trade-off between platform adoption and revenue accrual. That positive observation implies a normative corollary. If there is often little meaningful 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.²¹⁴ 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 the positive and the 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, can impose a net social loss under certain circumstances, and consistent with conventional assumptions (but confined to a far narrower ambit), can impose a net social gain under yet other circumstances. This Part preliminarily examines these possibilities in turn.

A. *The Indifference Baseline*

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 environment,²¹⁵ although the welfare of particular vendor populations may be substantially different.

²¹⁴ The normative consensus is not universal. Recent economic analysis expresses doubts regarding the net beneficial social welfare effects of OSS. See David S. Evans, *Politics and Programming: Government Preferences for Promoting Open Source Software*, in GOVERNMENT POLICY TOWARD OPEN SOURCE SOFTWARE 34 (Robert W. Hahn ed., 2002); Schmidt & Schnitzer, *supra* note 209; Engelhardt & Maurer, *supra* note 209; Maurer, *supra* note 109.

²¹⁵ Consistent with standard antitrust analysis, it is assumed that maximizing consumer welfare (in this case, equivalent to end-user welfare) is the selected policy objective. Some scholarly commentators take the view that antitrust should be concerned with total welfare. Both standards often yield the same outcome, although there can be important exceptions (as in mergers that confer productive efficiencies in excess of allocative inefficiencies). The debate is not relevant to my analysis. For further discussion, see Eleanor M. Fox, *The Efficiency Paradox*, in HOW THE CHICAGO SCHOOL OVERSHOT THE MARK: THE EFFECT OF CONSERVATIVE ECONOMIC ANALYSIS ON U.S. ANTITRUST 77, 78–79 (Robert Pitofsky ed., 2008).

This outcome 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. It is true that software may cost less if it is available through free download under an open source license rather than if it is available at some positive price from a proprietary vendor.²¹⁶ But, depending on competitive conditions in other segments of the consumption bundle, that may enable the hardware vendor (for example) to charge more — in fact, if the vendor dominates the hardware segment (and no firm has market power in any other segment), it can adjust pricing to extract exactly the amount the user has “saved” through the free download of software. In that simple scenario, no party has gained or lost except the software vendor (lost) and the hardware vendor (gained): user welfare remains constant. This case is stylized, but it illustrates a broader point. Arguments in favor of open platforms on grounds of public interest — the rhetoric favored by the explicit and implicit industry consortia that promote OSS adoption in the enterprise computing and smartphone markets — may sometimes reduce in part to privately interested arguments in favor of shifting rents from one particular set of industry players to another. There does not appear to be any 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 (Verizon) over integrated operating system and handset vendors (Apple) (as in the smartphone operating systems market). If that is the case, then the use of “open” or “closed” models for intellectual production may sometimes merely be a competitive choice bereft of policy implications that would invite state intervention.²¹⁷

²¹⁶ As a practical matter, “free” software is never really free. 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 if one were to take account of the total cost of ownership, especially given that maintenance costs are the far greater portion of those total costs. See Evans, *supra* note 214, at 42; LiMo Found., *supra* note 204, at 15–23. For evidence that the total cost of ownership of open source applications sometimes appears to exceed the proprietary alternative, see Schmidt & Schnitzer, *supra* note 209, at 496–97.

²¹⁷ More precisely, this Article refers 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 approach may simply be a form of protectionism: given the absence of any major European software provider with the exception of Germany’s SAP (which in turn may

B. The Case for Closed Models

It is even possible to suppose reasonable circumstances where the public interest would favor a closed model over an open model with respect to any individual component of any given consumption bundle. Suppose a market where firm *A* distributes software for free, either by choice or due to the lack of any legal or technological means by which to regulate access. Firm *A* must then secure a funding stream in order to cover its software development and maintenance costs; assume it can do so through sales of complementary hardware products. That scenario roughly describes the position of IBM in the mainframe market that preceded the advent of the market for personal computers and prepackaged software: like other hardware vendors, it bundled software with a hardware computing device and did not price the software separately.²¹⁸ Now suppose further that firm *A* has a dominant position in the hardware market (for example, IBM in the mainframe era) and firm *B* (for example, a startup called Microsoft) could develop a software product but has no immediate ability to produce a complementary hardware product.²¹⁹ As the stand-alone provider of a software product, firm *B* faces two unattractive entry opportunities into the market: (i) it can contract with firm *A*, which will most likely 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) fixed costs of developing the capacity to produce or otherwise deliver a complementary hardware or other product over which it can regulate access and earn positive revenues. These unfavorable options explain why no firm can easily enter the market today with *only* a browser technology²²⁰ (Google, Mozilla, and

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 foreign governmental policies favoring OSS, see Lee, *supra* note 209, at 56–59. Alternatively, one could argue that the relative paucity of U.S. government initiatives favoring OSS procurement reflects the political influence of the subset of proprietary firms that currently enjoy market success based on closed structural models.

²¹⁸ See FRANKLIN M. FISHER ET AL., *FOLDED, SPINDLED, AND MUTILATED: ECONOMIC ANALYSIS AND U.S. v. IBM* 9 (1983).

²¹⁹ 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 of equivalent quality at a lower cost than firm *B*.

²²⁰ One might reasonably object that neither obstacle constitutes an entry barrier — which is why this Article states that the startup faces “unattractive entry opportunities” rather than a simple entry barrier. Objections are as follows: (i) in the first case, firm *A* will rationally forfeit a share of joint revenues to firm *B* if doing so is required to elicit firm *B*’s investment (and if, even after such forfeiture, adding *B*’s product to the consumption bundle will result in a marginal net gain for *A*); and (ii) in the second case, external capital will fund *B*’s development costs if the project has a positive expected net present value. But each objection is in turn vulnerable to an

Microsoft supply it for free) or a search engine technology²²¹ (Google, Microsoft, and others supply it for free). Using an open model to distribute any given component in the total consumption bundle — in this stylized example, software — reduces competition in (in trade parlance, “commoditizes”) the market for that component by compelling any entrant to incur the costs of delivering some other complementary good or service in order to earn revenues.

This argument is substantially consistent with observed trends in the evolution of OSS development. As this Article has described in detail, the most widely distributed open source applications are now funded and partially governed by a handful of large ICT (mostly hardware) firms.²²² This movement is a necessary stage in the maturation of any open innovation environment: access must be regulated over some complementary good in order to satisfy the cost-recovery imperative. As is 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 dominates funding, governance, and staffing of leading open source projects in operating system development for the enterprise and mobile computing markets. Giveaways of software code accelerate adoption of a commoditized platform that promotes the economic fortunes of firms that have a competitive advantage in some other set of complementary assets — and conversely, they 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.²²³ IBM *again* gives away

important rejoinder: (i) in the first case, the objection ignores the high transaction costs inherent to contracting over yet-to-be-developed innovations and the holdup threat inherent to contracting over already developed innovations (for extensive discussion of these points, see generally Suzanne Scotchmer, *Standing on the Shoulders of Giants: Cumulative Research and the Patent Law*, J. ECON. PERSP., Winter 1991, at 29); and (ii) in the second case, it implausibly assumes perfectly efficient external capital markets (or at least, that the cost of external capital does not exceed the cost of internal capital). For a leading argument on why assuming perfectly efficient external capital markets is contestable (especially in the case of entry opportunities for which large capital investments are required), see Williamson, *supra* note 59, at 112.

²²¹ 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 fee.

²²² Professor Ronald Mann has similarly argued that the rise of OSS will support industry consolidation by privileging the holders of complementary assets that must be used to capture returns on unprotected assets. See Ronald J. Mann, *Commercializing Open Source Software: Do Property Rights Still Matter?*, 20 HARV. J.L. & TECH. 1, 32–33 (2006). Much of the evidence presented in this Article shows that these views have been further confirmed by the subsequent evolution of the OSS market.

²²³ I say “reinstates *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

software, for which it incurs hundreds of millions of dollars in development costs, in order to sell proprietary hardware (plus inherently excludable services and proprietary software), for which it presumably earns a revenue stream in an equal or greater amount. Put differently, controlled forfeiture of the operating system constitutes an alternative strategy by which to recover rents in a complementary goods market that were previously lost in direct competition over a proprietary position in the platform market.

That strategy may or may not be good for end users, who may be indifferent among states of the world that redistribute the aggregate rent pool among the holders of different portions 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 of the “IT stack,” who face higher entry costs given the de facto requirement to supply at least one other component in the stack in order to recover costs on the “free” software component.²²⁴ That inherent exclusionary effect casts a different light on the nonprofit consortia that provide competing operating systems for the smartphone market. Commoditizing 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.²²⁵

distributed by IBM in the mainframe era could run only on its hardware. That fact may considerably restrain the net anticompetitive effects of IBM's unilateral promotion of an open platform in order to advance sales of its complementary goods.

²²⁴ I say “at least some entrepreneurs” because smaller software firms can (and do) exploit commercial open source models. These models usually use a form of dual licensing in which a proprietary alternative offers greater features or support (or both); where successful, that strategy sometimes results in acquisition or investment by a larger proprietary firm that holds complementary assets through a larger bundle of products and services. On the use of modified open source business models (often involving commercial licensing) by startups and other commercial firms, see 451 GRP, *supra* note 107, at 1, 13–14, 16–17; and Philip H. Albert, *Dual Licensing: Having Your Cake and Eating It Too*, LINUXINSIDER (Nov. 16, 2004, 5:00 AM), <http://www.linuxinsider.com/story/38172.html>. For an example of a transaction involving the acquisition by a large proprietary firm of an open source software firm using a dual licensing strategy, see Larry Dignan, *Sun Acquires MySQL; Adds to Its Software Stack*, ZDNET (Jan. 16, 2008, 7:45 AM), <http://www.zdnet.com/blog/btl/sun-acquires-mysql-adds-to-its-software-stack/7611>, which describes the acquisition by Sun Microsystems, a large hardware and software manufacturer, of MySQL, an open source provider of database software, for approximately \$1 billion.

²²⁵ 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 section III.A.1, p. 1891. Given that commoditization effects can channel competition

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 commoditizing the platform technology promotes entry into 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 commoditization of the personal computer — a result that promoted the mass distribution of personal computing devices. But the former possibility cannot be excluded. Depending on competitive conditions in the remaining portions of the supply chain, commoditizing 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 commoditization of the hardware component, 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.

C. The Case for Open Models

The indeterminate outcome reached in the previous section may be analytically unsatisfying. Perhaps a different outcome results if one supposes that neither firm *A* in the previous example 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 commoditization 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 — assume that the consumption bundle is reduced to two components for simplicity — would allow any firm to capture anything more than a competitive return, firms could not recover their fixed R&D costs and would therefore decline to make further innovations. To garner a price

toward points on the consumption bundle at which a given firm has a competitive advantage (and away from other firms that do not), it may be argued that giveaway strategies may be explained on this basis alone. But the two explanations are at least nonexclusive and often complementary: the competition-channeling effects generate the revenue streams from complementary goods and services that render forfeiture an economically feasible commitment mechanism.

ing 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 outcome largely restores an indifference result: if firms cannot achieve pricing power in any portion of the consumption bundle, then either (i) commoditizing the software component would not result in any long-lasting improvement in consumer welfare (since the funding stream would be withdrawn and the component would be re-privatized) or (ii) no firm would make any effort to commoditize 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 *partially* consistent with the standard presumption favoring open over closed models. Suppose that a firm has a dominant position in the enterprise hardware market so that it will release complementary software at a reduced or zero price in order to enhance sales of the hardware product. Now suppose that there are two users: (i) enterprise user *A*, which must obtain both the software and the enterprise hardware to achieve its desired objective, and (ii) sophisticated individual user *B*, who must obtain the software but does not require the enterprise hardware to achieve her desired objective (for example, she can run the free software on commodity personal computing hardware). For user *A*, the indifference thesis proposed above holds: it pays roughly the same aggregate price across markets but simply allocates this price differently among firms. However, for user *B*, free software results in an aggregate reduction in her total price burden because she never pays the enterprise hardware provider: the enterprise user (who must purchase the complementary hardware product) is subsidizing use by the sophisticated personal user. In this case, free software still might not yield any aggregate efficiency gain, but it does yield a user-to-user distributional gain that may be relevant: namely, it enables the provider to charge a higher price to enterprise user *A*, which in turn subsidizes free distribution to sophisticated individual user *B*. Whether that distributional transfer — a far less dramatic social gain than the across-the-board reduction in access costs typically attributed to open innovation models — matters sufficiently (if at all) from a social perspective does not appear to give rise to any firm conclusions that would clearly call for state intervention.²²⁶ Even if that distributional

²²⁶ This argument explains why sophisticated individual users are the most zealous proponents of open source software while the vast remainder of the individual user population continues to use Windows. This Article leaves open the possibility that facilitating access by sophisticated users to

transfer *does* matter sufficiently from a social perspective, any minimally persuasive position in favor of state intervention would still require identifying impediments that would prevent the market from reaching the desired outcome independently. It is not clear that the last condition would be easily satisfied: as has been illustrated extensively, technology firms often face strong competitive pressures to open up access in order to attract user investments that are essential to sustaining platform value.

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 noneconomic factors as explanatory variables, overlook the use of access giveaways by commercial entities (which would be expected to select maximal control) and the use of access controls by ostensibly noncommercial entities (which would be expected to select no controls). 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 trade-off 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 trade-off 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 that 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 violates the insolvency constraint

a platform asset may generate social gains for all other users in the form of increased innovation (but not taking into account potentially offsetting social losses in the form of depressed innovation by stand-alone proprietary providers in the relevant platform market). Even in that case, however, platform hosts may have privately interested incentives to generate those social gains by expanding access for sophisticated users who can develop complementary applications. As described extensively throughout, platform holders in ICT markets have often adopted precisely that strategy.

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 of the Apache Software Foundation (Mar. 25, 1999), <http://www.apache.org/foundation/records/certificate.html>.

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

Apache License, Version 2.0 (Jan. 2004), <http://www.apache.org/licenses/LICENSE-2.0.html>.

GNOME Foundation

The GNOME Foundation Charter (Oct. 23, 2000), <http://foundation.gnome.org/about/charter>.

Bylaws of GNOME Foundation (Apr. 5, 2002), <http://foundation.gnome.org/about/bylaws.pdf>.

GNU Lesser General Public License, Version 2.1 (Feb. 1999), <http://www.gnu.org/licenses/lgpl-2.1.html>.

LiMo Foundation²²⁷

Bylaws of LiMo Foundation (Sept. 28, 2009), http://www.limofoundation.org/images/stories/pdf/090928_pub_limo_bylaws_consolidated_as_of_september_28th_09.pdf.

²²⁷ No license is included for the LiMo Foundation because, as provided in the LiMo Bylaws listed below, the Foundation offers its members the choice of various terms under which members may make contributions to the LiMo platform and other components. Some of those terms resemble open source licenses; others resemble conventional proprietary licenses. For further discussion, see Christopher Edwards, LiMo Found., LiMo Foundation Intellectual Property Policy (Dec. 3, 2010), http://www.limofoundation.org/images/stories/pdf/101203_con_limo_ipr_policy_final_v3%200.pdf; and *Open Source Policy*, LIMO FOUND., <http://opensource.limofoundation.org/index.php/open-source-policy.html> (last visited May 5, 2011).

Linaro Foundation²²⁸

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

Membership Rules of Linaro Limited,
<http://www.linaro.org/assets/PDF/MembershipRulesofLinaroLimited.pdf>.

Linux Foundation

Amended and Restated Bylaws of the Linux Foundation (Aug. 9, 2007), <http://www.linuxfoundation.org/about/bylaws>.

GNU General Public License, Version 2 (June 2001),
<http://www.gnu.org/licenses/gpl-2.0.html>.

Mozilla Foundation; Mozilla Corporation

Articles of Incorporation of Mozilla Foundation (July 14, 2003),
<http://www.mozilla.org/foundation/documents/mf-articles-of-incorporation.pdf>.

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

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

Mozilla Public License, Version 1.1,²²⁹ <http://www.mozilla.org/MPL/MPL-1.1.html>.

²²⁸ The Linaro Foundation has not adopted a standard licensing policy. However, it states: “[L]icensing used will be in line with the existing licensing plan for the open source projects.” *Linaro FAQ*, LINARO, <http://www.linaro.org/faqs> (last visited May 5, 2011).

²²⁹ Mozilla offers users two additional licensing options that impose greater constraints on proprietary derivative applications: (i) the GNU General Public License, version 2.0 or later, *GNU General Public License*, GNU OPERATING SYSTEM (June 1991), <http://www.gnu.org/licenses/gpl-2.0.html>; and (ii) the GNU Lesser General Public License, version 2.1 or later, *GNU Lesser General Public License*, GNU OPERATING SYSTEM (Feb. 1999), <http://www.gnu.org/licenses/lgpl-2.1.html>. See *Mozilla Foundation Licensing Policy, Version 2.1*, MOZILLA, <http://www.mozilla.org/MPL/license-policy.html> (last visited May 5, 2011).

Symbian Foundation²³⁰

Articles of Association of Symbian Foundation Limited (Feb. 26, 2009).

Memorandum of Association of Symbian Foundation Limited (Feb. 26, 2009).

Symbian Foundation Membership Rules (2009).

Eclipse Public License — Version 1.0, <http://www.eclipse.org/org/documents/epl-v10.php>.

Ubuntu Foundation²³¹

GNU General Public License, Version 2, <http://www.gnu.org/licenses/gpl-2.0.html>.²³²

²³⁰ As noted previously, the Symbian Foundation closed as of December 17, 2010, and transitioned to a licensing entity responsible for managing the Symbian trademark and other intellectual property. See *supra* note 191 and accompanying text. The Symbian Foundation documents are no longer available online; copies of the instruments listed above are on file with the Harvard Law School Library.

²³¹ No formal governance documents for the Ubuntu Foundation are listed because none are available on the Foundation's website or, it seems, through other sources. All descriptions of the governance structure of the Foundation reflect information provided on the Foundation's website. See *About Ubuntu Governance*, UBUNTU, <http://www.ubuntu.com/project/about-ubuntu/governance> (last visited May 5, 2011); *CommunityCouncil*, UBUNTU WIKI, <https://wiki.ubuntu.com/CommunityCouncil> (last visited May 5, 2011).

²³² The GPL is the most commonly used license for software releases by the Ubuntu Foundation. Some software may be released under other licenses; however, with the exception of certain proprietary hardware drivers, all such licenses must comply with certain open source requirements. For further discussion, see *About Ubuntu Licensing*, UBUNTU, <http://www.ubuntu.com/project/about-ubuntu/licensing> (last visited May 5, 2011).