Harvard Journal of Law & Technology
Volume 29, Number 1 Fall 2015

PREDATORY INNOVATION IN SOFTWARE MARKETS

Suzanne Van Arsdale & Cody Venzke[*]

TABLE OF CONTENTS

[*] Suzanne Van Arsdale is an associate with Sidley Austin LLP. Suzanne earned a J.D. from Harvard Law School in 2015 and three degrees from Clarkson University in 2010: a B.S. in Biomolecular Science, a B.S. in Biology, and a B.S. in Psychology.

Cody Venzke is an associate with Paul Hastings LLP. Cody earned a J.D. from Stanford Law School in 2015 and a B.A. from St. Olaf College in 2010.

The views expressed in this article are exclusively those of the authors and do not necessarily reflect those of Sidley Austin LLP and its partners, nor of Paul Hastings LLP and its partners. This Note has been prepared for informational purposes only and does not constitute legal advice. This information is not intended to create, and receipt of it does not constitute, a lawyer-client relationship. Readers should not act upon this without seeking advice from professional advisers.

The authors wish to thank Professors Mark Lemley and Louis Kaplow for advising the paper that inspired this Note, Zach Shepherd for his technical expertise, Peter Arnaboldi for his helpful comments as Article Editor, and the editors and staff of the *Harvard Journal of Law & Technology* for their dedication and support.

I. INTRODUCTION

At its core, antitrust doctrine seeks to promote competition and, in turn, innovation. In theory, monopolists may run afoul of the Sherman Act[1] by altering their products to prevent competitors from entering the market — a violation of antitrust law known as predatory innovation. In practice, courts have refused to find liability as long as the market has accepted the changed product, “innovative” or not. This doctrine, which has generally been accepted as well-grounded, developed during a period predating the rise of the Internet. Consequently, the doctrine of predatory innovation (or rather, the one against it) depended on implicit assumptions about products that were primarily physical in nature.

However, the rise of computing technologies and the emergence of the Internet have created a market in software products which defies these assumptions. This Note defines software to encompass all code-based, machine-intelligible instructions, including any non-physical components of a computer system, such as operating systems, applications, websites, and social media platforms. Unlike many physical products subject to antitrust allegations stemming from predatory innovation, software products, especially web services, (1) are more

[1] 15 U.S.C. § 2 (2012).

likely to depend on network effects for success; (2) allow producers to effectively discontinue availability of and support for older products; and (3) can be changed with minimal or no effect on the user experience.

Consequently, a dominant firm may harm competition by limiting the interoperability, or compatibility, of a dominant product with those of competitors. Interoperability is the ability of software to interact with other software or hardware by (1) exchanging information and (2) using the exchanged information; if software cannot interoperate with a depended-upon component, it may lose features or not work at all. Because of the nature of software products, breaks in interoperability may occur without users ever being aware that they are dealing with a new version of the product. This new software market calls for a reexamination of the applicability of existing predatory innovation doctrine to software products.

This Note lays the foundation for such a reexamination. Part II outlines the type of conduct we see as problematic: breaks in formerly interoperable software products. This conduct, we argue, is of particular concern to antitrust analysis as it threatens to hurt competition in software markets for no purpose other than the exclusion of competition. Part III traces the law of predatory innovation as it currently stands, including the assumptions courts made about products as they developed the doctrine. In particular, this Part finds that markets of traditional, physical products generally satisfy the three assumptions mentioned above, that (1) individual products do not depend on network effects; (2) producers and consumers may continue to sell older products even as new iterations enter the market; and (3) the market is able to recognize when a product change is made and respond to that change in choosing among competing products. Part IV examines the nature of software markets and how they differ for those that gave rise to predatory innovation law. This Part argues that software products do not adhere to the three assumptions that underlie the development of predatory innovation doctrine; consequently the application of the doctrine as it now stands to software products is uneasy at best.

Part V rejects the applicability of other traditional tests for monopolization, such as tying and unilateral refusals to deal, in the software context. As discussed below, tying cannot reach many instances of predatory innovation in software products, as the anticompetitive behavior does not necessarily include two distinct products. Likewise, tests premised on *Aspen Skiing Co. v. Aspen Highlands Skiing Corp.* may not apply without some prior contractual agreement between the parties. The essential facilities doctrine may not apply as software markets lack the limitations on reproducibility that are more readily apparent in physical facilities. Finally, Part VI articulates a new structured rule of reason test for predatory innovation in software markets.

The test sequentially assesses whether there was an explicit or implied commitment to support specific or general interoperability, the timing of the break in interoperability relative to other changes, the necessity or reasonableness of non-interoperability to enable those other changes in the product, and whether the other changes were genuine innovations accepted by the market.

II. Breaks in Interoperability Among Previously Compatible Products Pose the Chief Threat to Competition in Markets of Software Products

The conduct at the heart of this Note is breaks in interoperability in software products. Interoperability, as described below, refers to the ability to (1) exchange information and (2) use the exchanged information.[2] There are some procompetitive grounds for restricting interoperability, but interoperability can be beneficial for an industry and consumers,[3] particularly when it involves use of an industry standard.

Unfettered interoperability is not always practical or desirable for individual competing firms. Permitting restrictions on interoperability — through technical constraints, such as manipulating mandatory or negative dependencies with other products, and through legal constraints, such as intellectual property enforcement rights — can drive innovation and technological progress. Software products, like physical products, are valuable and costly to develop, and intellectual property rights and the ability to restrict access can provide valuable incentives for firms to invent, develop, and improve those products. Additionally, interoperability can be costly and difficult to achieve, especially if the underlying product must work with a range of other products.[4] Mandated interoperability can reduce innovation; if new entrants cannot prevent other products from interacting with or using the same data, then consumers may take that data and switch to a competing product, and the firm may lose other potential first-mover benefits.[5]

[2] *See infra* IV.A.2.

[3] *See* Peter Swire & Yianni Lagos, *Why the Right to Data Portability Likely Reduces Consumer Welfare: Antitrust and Privacy Critique*, 72 MD. L. REV. 335, 376–79 (2013).

[4] *Id.* at 355–56 (citing Rajiv Shah & Jay P. Kesan, *Lost in Translation: Interoperability Issues for Open Standards*, 8 I/S: J.L. & POL'Y FOR THE INFORMATION SOC'Y 113, 113 (2012), http://moritzlaw.osu.edu/students/groups/is/files/2012/02/Kesan.pdf [http://perma.cc/TGE9-D2E5]) (describing a study of problems with interoperability in several open file formats).

[5] *Cf.* ANDREJ FATUR, EU COMPETITION LAW AND THE INFORMATION AND COMMUNICATION TECHNOLOGY NETWORK INDUSTRIES: ECONOMIC VERSUS LEGAL CONCEPTS IN PURSUIT OF (CONSUMER) WELFARE 81 (2012) ("The right to exclude ensures that successful innovators can recover their sunk costs and receive a return that compensates them for the risk.").

However, when a software product achieves market dominance, subsequent limitations on interoperability may become problematic,[6] as the value of a software product may lie in its popularity rather than its technical merit.[7] For example, for social media platforms and software in which user interaction and network effects are a key part of the product's value proposition, value depends heavily on the number of users.[8] This creates an opportunity for established, dominant firms to take particular advantage of breaks in interoperability, as severing interoperability may impact the smaller network much more dramatically.

A. Breaks in Interoperability in Practice: Three Examples

Recent developments in web-based platforms demonstrate the potential anticompetitive effects of breaking interoperability. Numerous firms have broken interoperability in public and controversial ways, often by changing their application programming interface ("API")[9] or other software to remove support for complementary products by competing third parties. Examples include recent conduct by Apple, Twitter, and Facebook. We examine the history of each before turning to their potential anticompetitive effects.

In 2004, RealNetworks ("Real") released its Harmony technology to allow songs sold through its music store to play on Apple's market-

[6] Peter Lee, Response, *Innovating Between and Within Technological Paradigms: A Response to Samuelson*, 94 MINN. L. REV. HEADNOTES 1, 6–10 (2009) ("When an industry has coalesced around a particular standard, the interfaces underlying that standard approach the status of 'infrastructure,' which economic theory suggests should be widely accessible . . . . [W]hen a patented interface becomes an industry-wide standard, relaxing exclusive rights may be warranted to promote 'normal' progress within the paradigm." (citations omitted)); Brett M. Frischmann, *An Economic Theory of Infrastructure and Commons Management*, 89 MINN. L. REV. 917, 922–23 (2005) ("[I]f a resource can be classified as infrastructure . . . there are strong economic arguments that the resource should be managed in an openly accessible manner"); Peter Lee, *The Evolution of Intellectual Infrastructure*, 83 WASH. L. REV. 39, 84 (2008) ("[E]quating commonplace entities with nonprotectable ideas ensures that widely used infrastructure remains in the public domain.").

[7] *See* Mark A. Lemley & David McGowan, *Legal Implications of Network Economic Effects*, 86 CALIF. L. REV. 479, 497 (1998); *see also* Jacob Kastrenakes, *Instagram Is Now Valued at $35 Billion*, VERGE (Dec. 19, 2014), http://www.theverge.com/2014/12/19/7420981/instagram-valued-at-35-billion [http://perma.cc/QM98-EQZE] ("The increased value [of the company] is due to Instagram's announcement last week that it now has over 300 million active monthly users — meaning that it's growing faster than expected.").

[8] Metcalfe's law of network effects (n^2) was the basis for the Facebook IPO valuation, and others have proposed the less optimistic Zipf's law (n log(n)) as an alternative. *See* Anthony Wing Kosner, *Facebook Values Itself Based on Metcalfe's Law, But the Market Is Using Zipf's*, FORBES (May 31, 2012, 1:14 PM), http://www.forbes.com/sites/anthonykosner/2012/05/31/facebook-values-itself-based-on-metcalfes-law-but-the-market-is-using-zipfs [http://perma.cc/8MXH-E63H].

[9] For a technical explanation of interoperability and APIs, see *infra* Part IV.A.2.

dominant iPods;[10] days after Real's release of Harmony, Apple announced that it would update iTunes and its digital rights management ("DRM") technology to prevent songs sold by competitors, such as Real, from working on iPods.[11] After it did just that, plaintiffs filed suit, alleging that Apple's conduct was exclusionary and constituted illegal tying. Although Apple maintained that the DRM technology served to placate record labels,[12] Real spun off its music store in 2010;[13] in 2007, Apple CEO Steve Jobs admitted that DRM had done little to prevent music piracy.[14]

Facebook, Google, and other social networking websites often restrict data portability by limiting or blocking the ability of users and other websites to access user information or export it to another website.[15] This breaks interoperability by preventing information exchange. Twitter, for example, has limited the number of users that may access its network from third-party mobile applications.[16] This limitation effectively caps the market share of users that any third-party developer can capture — forcing new users to adopt Twitter's proprietary mobile application.[17] Twitter claims that the changes offer users an improved, unified experience across platforms,[18] but Twitter's own mobile application has routinely received poorer reviews

[10] Micah Singleton & Josh Lowensohn, *Apple's DRM Lawsuit: 10 Years in the Making*, VERGE (Dec. 4, 2014, 5:17 PM), http://www.theverge.com/2014/12/4/7333609/apples-drm-lawsuit-10-years-in-the-making [http://perma.cc/NCA2-TFXG].

[11] Joe Mullin, *Apple Will Face $350M Trial over iPod DRM*, ARS TECHNICA (Oct. 3, 2014, 12:05 PM), http://arstechnica.com/tech-policy/2014/10/apple-will-face-350m-trial-over-ipod-drm [http://perma.cc/98K3-SW7R].

[12] Singleton & Lowensohn, *supra* note 10.

[13] Kelsey Swanekamp, *RealNetworks Spins Off Rhapsody*, FORBES (Feb. 10, 2010), http://www.forbes.com/2010/02/10/realnetworks-rhapsody-music-markets-equities-closer.html [http://perma.cc/P2EC-7TKM].

[14] Mike Musgrove, *Jobs Calls for Open Music Sales*, WASH. POST (Feb. 7, 2007), http://www.washingtonpost.com/wp-dyn/content/article/2007/02/06/AR2007020601764.html [http://perma.cc/V48G-87R3].

[15] *See, e.g.*, Alexei Oreskovic, *Google Bars Data from Facebook as Rivalry Heats Up*, REUTERS (Nov. 5, 2010, 7:06 PM), http://www.reuters.com/article/2010/11/05/us-google-facebook-idUSTRE6A455420101105 [http://perma.cc/C3ZF-RHQQ]; *see also* David Gelles, *Facebook Accused of Restricting Its Users*, FIN. TIMES (July 11, 2009, 1:01 AM), http://www.ft.com/intl/cms/s/2/82860a80-6da1-11de-8b19-00144feabdc0.html [http://perma.cc/6TZR-L6Z9] (describing the dispute between Facebook and rival Power Integrations over restrictions on scraping data off Facebook).

[16] *See* Ryan Lawler, *Twitter Gives Devs 6 Months To Display Tweets Properly, Use New Authentication and Rate Limits*, TECHCRUNCH (Aug. 16, 2012), http://techcrunch.com/2012/08/16/twitter-gives-developers-6-months-to-properly-display-tweet-use-new-authentication-and-rate-limts [http://perma.cc/8LX5-B4SF].

[17] *See* Anthony Ha, *Twitter Handcuffs Clients Apps with New API Changes*, TECHCRUNCH (Aug. 16, 2012), http://techcrunch.com/2012/08/16/twitter-api-client-apps [http://perma.cc/QT4U-RHMH] (arguing that Twitter's API changes are to limit third-party applications' shares of the user base).

[18] *See* Lawler, *supra* note 16 (detailing Twitter's announcement, especially limitations on calls for content and number of user tokens permitted).

than its third-party competitors in various app stores.[19] Twitter's restrictions do not appear to further the user experience, and may serve a primarily anticompetitive purpose by limiting rival mobile applications' access to users and, in turn, ad revenue.[20]

Likewise, in 2012, Facebook acquired the photo sharing startup Instagram.[21] Prior to the acquisition, Instagram users had been able to share photos across a variety of social networks, including the microblogging platform — and Facebook rival — Twitter.[22] Following the acquisition, however, Facebook altered Instagram's API, denying users the ability to cross-post photos to Twitter — in effect, preventing Twitter from accessing content on Instagram.[23] The arrangement had previously been beneficial to Instagram, as it expanded the reach of its users' content. Under the umbrella of Facebook's much larger social network, Instagram's interoperability was altered. Although Facebook may again point to improving its web experience,[24] the frustration of users suggests that the change ran against market preferences.[25]

In addition to running against user preferences, breaks in interoperability also pose threats traditionally recognized by antitrust law, which we turn to now.

[19] *See, e.g.*, Phil Nickinson, *Twitter's Android App Is Now Nearly Unusable: Try These Instead*, ANDROID CENTRAL (Dec. 11, 2013), http://www.androidcentral.com/new-twitter-app-android-alternatives [http://perma.cc/H2US-65YF].

[20] Xochi Adame, *Twitter's Antisocial Crackdown: A #Fail for Social Biz?*, CDS GLOBAL (Aug. 24, 2012), http://www.cds-global.com/blog/twitters-antisocial-api-crackdown-a-fail-for-social-biz/ [http://perma.cc/8PY4-XR86] ("As Ben Brooks has noted in his blog, The Brooks Review, 'Twitter was built as a community with users trying to improve the service the best they could for everyone's benefit — and that is now gone.'"); Matthew Panzarino, *Twitter Wants a Billion Users and It's Prepared to Sacrifice Developers to Get There*, TNW NEWS (Aug. 21, 2012), http://thenextweb.com/twitter/2012/08/21/twitter-secured/ [http://perma.cc/Z2DY-YNNN] ("I think that it's important, however, to make something clear: Beyond anything in development right now, you will never be getting great consumer-oriented third-party Twitter clients again.").

[21] Shayndi Raice & Spencer E. Ante, *Insta-Rich: $1 Billion for Instagram*, WALL ST. J. (April 10, 2012, 3:55 AM), http://www.wsj.com/articles/SB10001424052702303815404577333840377381670 [http://perma.cc/8HGE-5T6M].

[22] *See* Leena Rao, *Instagram Photos Will No Longer Appear in Twitter Streams at All*, TECHCRUNCH (Dec. 9, 2012), http://techcrunch.com/2012/12/09/it-appears-that-instagram-photos-arent-showing-up-in-twitter-streams-at-all [http://perma.cc/8FJG-CKNC] (explaining that the changes were intended to drive more traffic to Instagram's newly revamped web application).

[23] *See id.*

[24] *See id.*

[25] *See id.*; *see also* Alex Masters, *Instagram Cuts Off Twitter Card Support, But It's More Business than Personal*, INDEP. (Dec. 6, 2012), http://www.independent.co.uk/voices/comment/instagram-cuts-off-twitter-card-support-but-it-s-more-business-than-personal-8389884.html [http://perma.cc/P4D5-WFGK] ("[T]he seamless experience between social networks will continue to degrade as moneti[z]ation becomes increasing [sic] more important than the overall user experience.").

B. Threat of Exclusionary Conduct at Its Height

A product redesign that breaks interoperability with one or more products, such as by restricting access to or support for an interface, may signal anticompetitive effect. As described below,[26] users and competitors may come to rely on the software or content provided by a firm, especially in its product's infancy.[27] The firm may also encourage reliance by officially supporting an API or encouraging complementary development.[28] The resulting expanded reach may benefit the nascent monopolist, who could then exclude competitors from interoperating with the underlying product. The same network effects the firm relied upon to expand its user base then act as a barrier to entry, allowing little opportunity or motive for competitors to create their own products. In *Aspen Skiing Co. v. Aspen Highlands Skiing Corp.*, the Supreme Court ruled similar conduct — eliminating a profitable business deal with a rival after becoming a dominant firm — could very well serve to eliminate competition.[29] Here, the conduct by Twitter and Facebook seems to fit this paradigm. In both cases, the firm severed compatibility after reaching dominant status, despite previously supporting and benefitting from interoperability.[30]

If, however, a firm has no policy of interoperability — it has not opened its API and does not support complementary products officially or by way of encouragement — then a product change does not suggest anticompetitive conduct. The firm has created no reliance on the part of competitors and likewise has not benefitted from contributions made by competitors who relied on interoperability.

C. Countervailing Interests at Their Lowest

Antitrust law has long sought to accommodate interests running counter to a policy of free competition, including firms' freedom of association. Nonetheless, the countervailing interests of software producers are at their lowest when firms eliminate interoperability. In particular, antitrust law has a history of viewing the elimination of cooperative, mutually beneficial arrangements with suspicion. Cases such as *Lorain Journal Co. v. United States* and *Aspen Skiing* arguably establish that a First Amendment right of association gives way to

[26] *See infra* notes 120–26 and accompanying text.

[27] *See* James B. Speta, *Tying, Essential Facilities, and Network Externalities: A Comment on Piraino*, 93 NW. U. L. REV. 1277, 1281 (1999) ("[M]onopoly owners of a network good may have additional incentives not to preclude competitive offerings of complementary goods, especially in early phases of the development of the network, because consumers value bigger networks more highly.").

[28] *See infra* notes 120–26 and accompanying text.

[29] Aspen Skiing Co. v. Aspen Highlands Skiing Corp., 472 U.S. 585, 608 (1985).

[30] *See* Lawler, *supra* note 16; Masters, *supra* note 25.

congressional regulation where disassociation is predominantly anticompetitive. In *Lorain Journal*, the Court considered whether a monopolist newspaper could lawfully refuse to run advertisements from businesses that also advertised with a rival newspaper. The Court held that such a refusal was anticompetitive and therefore unlawful.[31] The monopolist newspaper had longstanding control of local and national advertisements in the city.[32] The Court reasoned that, although a business normally retains the right to choose customers or suppliers, that right is limited by the statutory rights of others.[33] In that case, the monopolist newspaper's right of association was limited by Congress's power under the Commerce Clause and the rights it had subsequently given consumers and competitors to market competition.[34]

As described above, the *Aspen Skiing* Court similarly considered whether a dominant firm could legally terminate a long-standing, profitable arrangement with rival ski areas.[35] Noting that the elimination of a profitable business agreement evinced an anticompetitive purpose, the Court easily dispensed with the monopolist's freedom of association claim by citing *Lorain Journal*.[36] Thus, where courts find that elimination of formally profitable cooperation with rivals reveals anticompetitive intent, they may hold that countervailing freedom of association concerns give way to Congress's regulation of interstate commerce under the Commerce Clause.

D. Standard-Setting Organizations Are Inadequate Regulators

Network effects and other market incentives may be strong enough to encourage interoperability, at least early on. Standard-setting organizations ("SSOs"), which adopt shared standards to increase business opportunity, further encourage interoperability among member firms by adopting standards licensed to members or made available to all with an open and permissive license.[37]

To ensure that members are able to use a standard if adopted, an SSO may require those participating in setting standards to disclose any proprietary interest in intellectual property, which might be enforced against users.[38] After adoption, however, an SSO may not have penalties, contractual leverage, or other enforcement mechanisms to prevent a firm that fails to make such a disclosure from subsequently

[31] Lorain Journal Co. v. United States, 342 U.S. 143, 147–49, 155 (1951).

[32] *See id.* at 147.

[33] *Id.* at 155.

[34] *See id.* at 155–56.

[35] *Aspen Skiing*, 472 U.S. at 604–05.

[36] *Id.* at 601–02.

[37] *See Standard-setting Organization*, PRACTICAL L., http://us.practicallaw.com/9-557-1858 [http://perma.cc/FMR9-2QDZ]; Pamela Samuelson, *Are Patents on Interfaces Impeding Interoperability?*, 93 MINN. L. REV. 1943, 1945 (2008).

[38] *See* Samuelson, *supra* note 37, at 2002–03.

enforcing its right to exclude against members.[39] Instead, members may allege anticompetitive conduct in the form of misrepresentation or fraud during the standard-setting process — due to omission, such as failure to assert ownership, or affirmative falsehood, asserting that no intellectual property rights are owned.[40] This antitrust claim of attempted monopolization under section 2 of the Sherman Act[41] is premised on manipulation of the standard-setting process in order to achieve market power. When the SSO adopts a standard known to be proprietary, the antitrust claim is weaker.[42]

An SSO may also require member commitments to license future intellectual property that uses adopted standards, to the extent necessary for interoperability, on royalty-free or reasonable and nondiscriminatory terms.[43] A firm using standards made available with an open and permissive license may extend them in a product with proprietary capabilities, and then use those differences and intellectual property enforcement rights to disadvantage competitors.[44] Once a firm has a sufficient number of users, it may change its product or policies to no longer be interoperable and thereby exclude competitors. This may be done in a variety of ways, such as extending or modifying the standards as used in their product or denying competitors access to content. However, once a dominant firm breaks interoperability, the SSO may again be powerless to enforce compliance or otherwise mitigate consequences to competitors.

The anticompetitive harms inflicted by predatory innovation in the cases of Apple, Twitter, and Facebook both limit free consumer choice and pose barriers to entry by competitors. Some predatory designs, such as Apple's manipulation of iTunes and DRM, parallel the

[39] *See* Deborah A. Coleman, *IP Risks Associated with Participating in Standard Setting Organizations*, HAHN LOESER & PARKS LLP (2009), http://www.hahnlaw.com/wp-content/uploads/2015/02/1016.pdf [http://perma.cc/M5S9-S6TF].

[40] In these circumstances, the court may also prohibit the rights holder from enforcing the patent. *See* Qualcomm Inc. v. Broadcom Corp., 548 F.3d 1004, 1025 (Fed. Cir. 2008).

[41] 15 U.S.C. § 2 (2012).

[42] Mark A. Lemley, *Intellectual Property Rights and Standard-Setting Organizations*, 90 CAL. L. REV. 1889, 1927–35 (2002) (discussing antitrust theories of liability).

[43] Antitrust concerns naturally arise in standard-setting organization activities, where competitors within the industry collectively discuss agreements to license. *See, e.g.*, Mark A. Lemley & Carl Shapiro, *Patent Holdup and Royalty Stacking*, 85 TEX. L. REV. 1991, 2042–43 (2007) (suggesting that royalty caps or "step-down" royalties may also be procompetitive but raise additional "antitrust flags"). For examples of SSOs, see Samuelson, *supra* note 37, at 1946–54.

[44] This problem can be avoided if the standards are available under a copyleft license — a license conditioned on derivative works being available under the same license terms. *See, e.g.*, *GNU General Public License*, GNU (June 2007), https://www.gnu.org/licenses/gpl.html [http://perma.cc/DP25-HKGA]. It can also be sidestepped if members are required to adopt a defensive patent license. *See generally* Jason Schultz & Jennifer M. Urban, *Protecting Open Innovation: The Defensive Patent License as a New Approach to Patent Threats, Transaction Costs, and Tactical Disarmament*, 26 HARV. J.L. & TECH. 1 (2012), http://jolt.law.harvard.edu/articles/pdf/v26/26HarvJLTech1.pdf [http://perma.cc/6BD8-47AJ] (proposing defensive patent license pools).

dangers of a tying arrangement, as the software changes come close to tying two products together and discourage consumers from considering competitors.[45] Other software design changes fit better under the rubric of unilateral refusals to deal, as the foreclosure in consumer choice lies in limiting competitors' access to or replication of content.[46] This conduct, however, does not fit neatly under the traditional tests for monopolization. After analyzing the assumptions underlying current predatory innovation doctrine, we demonstrate that both the tying and unilateral refusal to deal analyses are insufficient to address the above anticompetitive conduct.

III. THE CURRENT LAW OF PREDATORY INNOVATION

In the United States, a monopoly is not in and of itself illegal. Although the Sherman Act encourages free competition, courts have found it implausible that Congress sought to prohibit all monopolies, especially those that "won" in fair competition on the merits.[47] Thus, to face liability under antitrust law, a monopolist must (1) possess monopoly power in the relevant market, the ability to raise prices or reduce output without competitors entering the market, and (2) commit some anticompetitive conduct in furtherance of that monopoly.[48] The conduct element usually focuses on exclusionary behavior — efforts by the monopolist to keep competitors out.[49] Anticompetitive conduct runs the gamut of behavior, from requiring or encouraging customers to buy multiple products ("tying")[50] and pricing below cost ("predatory pricing") to refusing to deal or forcing suppliers or buyers to refuse to deal with a competitor.

Many cases involving software products, such as operating systems and software, depend on some notion of tying. A "technological tie-in" occurs when a company designs a product in a way that encourages or forces consumers to purchase another product from the

[45] *See, e.g.*, Digidyne Corp. v. Data Gen. Corp., 734 F.2d 1336, 1343 (9th Cir. 1984).

[46] *See, e.g.*, Data Gen. Corp. v. Grumman Sys. Support Corp., 36 F.3d 1147, 1183 (1st Cir. 1994).

[47] *See* Berkey Photo, Inc. v. Eastman Kodak Co., 603 F.2d 263, 275 (2d Cir. 1979) ("The mere possession of monopoly power does not ipso facto condemn a market participant.").

[48] United States v. Grinnell Corp., 384 U.S. 563, 570–71 (1966) ("[Section] 2 of the Sherman Act has two elements: (1) the possession of monopoly power in the relevant market and (2) the willful acquisition or maintenance of that power as distinguished from growth or development as a consequence of a superior product, business acumen, or historic accident.").

[49] *Id.* at 570–71, 576.

[50] Although technically a claim of collusive conduct under section 1 of the Sherman Act, tying requires a showing of market power, Jefferson Par. Hosp. Dist. No. 2 v. Hyde, 466 U.S. 2, 13–14 (1984), and thus throughout this Note, we treat it as closely related to monopolization under section 2 of the Act.

company rather than its competitors.[51] Examples include designing computer terminals so that they are compatible with peripheral devices only from the same company. Courts, however, have generally refused to find tying in software products, noting that the competitors may adapt to the technological tie-in over time by reverse engineering their peripheral products to be compatible with the dominant tying product.[52] The dominant firm, then, has not gained a monopoly in the tied peripheral market, only a head start.

In response, plaintiffs began raising the related claim of predatory innovation, alleging that a dominant firm had altered its product, not to tie with a peripheral one, but to eliminate interoperability with competitors' products or to create unjustified demand in the market. We now turn to the contours of the theory of predatory innovation.

A. Predatory Innovation Tests: Business Justification and Market Acceptance

Predatory innovation is a product change that does not advance competition on the merits and prevents competitors from efficiently entering the field.[53] In *C.R. Bard, Inc. v. M3 Sys., Inc.*, for example, the Federal Circuit considered whether changes in a firing gun for biopsy needles constituted predatory innovation.[54] In finding for the plaintiff, the court noted that a claim of predatory innovation required the plaintiff to show that the change was made for "predatory reasons, *i.e.*, for the purpose of injuring competitors in the replacement needle market, rather than for improving the operation of the gun."[55] In litigation over IBM's peripheral devices, the Northern District of California phrased the elements of the claim similarly. There the court stated that where the design changes have "had no purpose and effect other than the preclusion of [competitors]," they constitute predatory

[51] *See generally* Daniel E. Gaynor, *Technological Tying* (FTC, Working Paper No. 284, 2006), *available at* http://link.multcolib.org/portal/Technological-tying--Technological-tying/ExTe1EVd06U/ [http://perma.cc/4LAZ-8YN3].

[52] *See, e.g.*, Somers v. Apple, Inc. (*In re* Apple iPod iTunes Anti-Trust Litig.), Nos. C 05-00037 JW, C 07-06507 JW, 2010 WL 2629907, at *3 (N.D. Cal. June 29, 2010) ("Plaintiffs must allege facts showing that Apple engaged in anticompetitive conduct, with the specific intent to control prices or destroy competition, beyond the technological interoperability of iPods and media sold through the iTS."); Foremost Pro Color, Inc. v. Eastman Kodak Co., 703 F.2d 534, 542 (9th Cir. 1983) ("We do not believe that, standing alone, such technological interrelationship among complementary products is sufficient to establish the coercion essential to a *per se* unlawful tying arrangement."); HDC Med., Inc. v. Minntech Corp., 411 F. Supp. 2d 1096, 1101 (D. Minn. 2006) (collecting cases).

[53] *See* C.R. Bard, Inc. v. M3 Sys., Inc., 157 F.3d 1340, 1382 (Fed. Cir. 1998) ("In order to prevail on its claim . . . M3 was required to prove that Bard made a change in its Biopty gun for predatory reasons, i.e., for the purpose of injuring competitors in the replacement needle market, rather than for improving the operation of the gun.").

[54] *Id.* at 1346.

[55] *Id.* at 1382.

innovation.[56] The scope of antitrust liability for predatory innovation is broader than for tying claims; there is no need for a second, tied product, or for a contractual relationship between the parties — only for the monopolist to change the product at issue.

Despite, or perhaps because of, the apparent breadth of the predatory innovation doctrine, courts have been reluctant to embrace it; if anything, its rejection has been more notable than its acceptance. Courts' hesitance to accept predatory innovation claims has been spurred by at least two concerns. First, courts are reluctant to chill innovation: it plays a central role in competition, particularly non-price competition, and is key not only to a functioning free market, but also to the public policies enshrined in the Sherman Act.[57] Consequently, defendants carry only the minimal burden of providing any business justification; any minor improvement suffices for the defendant to prevail, and concurrent changes resulting in incompatibility can be justified as accommodating that "improvement."[58] Second, courts may be skeptical of their own competence to assess whether software product changes are genuinely innovative.[59] As a result of these two concerns, courts defer to defendants who can show some minor improvement or business justification, even if the change otherwise has anticompetitive effects.

As a result of this reluctance to chill innovation or operate beyond their expertise, courts have adopted at least two tests to assess predatory innovation. Although the language of court opinions echoes the classic rule of reason,[60] the tests employed by courts generally do not rely on a balancing of harms as one might expect under such a standard.[61] Instead, courts have relied primarily on either an intent-based

[56] *In re* IBM Peripheral EDP Devices Antitrust Litig., 481 F. Supp. 965, 1002–03 (N.D. Cal. 1979) (articulating four factors for evaluating the business merit of the changes).

[57] *See Berkey Photo*, 603 F.2d at 286 ("The attempt to develop superior products is, as we have explained, an essential element of lawful competition. Kodak could not have violated § 2 merely by introducing the 110 camera with an improved film.").

[58] *See* Expert Masonry, Inc. v. Boone Cty., 440 F.3d 336, 348 (6th Cir. 2006) (quoting Crane & Shovel Sales Corp. v. Bucyrus-Erie Co., 854 F.2d 802, 809 (6th Cir. 1988)) (noting courts' general deference to business decisions and finding predatory innovation only under a no-business-sense test); United States v. Microsoft Corp., 253 F.3d 34, 65 (D.C. Cir. 2001) (noting courts' skepticism toward claims based on product design changes by the dominant firm); Oahu Gas Serv., Inc. v. Pacific Res., Inc., 838 F.2d 360, 369 (9th Cir. 1988) (noting courts' consistent rejection of the "predatory innovation" theory).

[59] *Berkey Photo*, 603 F.2d at 287 ("[N]o one can determine with any reasonable assurance whether one product is 'superior' to another . . . ."); *see also* United States v. Microsoft Corp., 147 F.3d 935, 953 (D.C. Cir. 1998) (noting the undesirability of having courts oversee product design); *In re IBM*, 481 F. Supp. at 1003 (citing ILC Peripherals v. Int'l Bus. Machs., 458 F. Supp. 423, 439 (N.D. Cal. 1978)) ("[W]here there is a valid engineering dispute over a product's superiority the inquiry should end; the product is innovative and the design is legal.").

[60] Abbott Labs. v. Teva Pharm. USA, Inc., 432 F. Supp. 2d 408, 422 (D. Del. 2006) ("Hence, an inquiry . . . following the rule of reason approach, is justified.").

[61] *See Microsoft*, 253 F.3d at 59, 66–67 (including balancing as the last step of its predatory innovation test but not applying that step, presumably because the defendant had

legitimate business justification test[62] or a market acceptance test,[63] with variations depending on the circuit.[64] In *Berkey Photo v. Eastman Kodak Co.*, for example, the Second Circuit expressly dismissed a predatory innovation cause of action where the market had accepted the new product.[65] The court's rationale seemed to recognize not only its own limitations as an arbiter of innovation but also that the notion of improvement or innovation was inherently vague. Only the market could ultimately determine which products were improvements on their predecessors.[66]

The Federal Circuit, however, has hewed more closely to a test that relies on intent.[67] Rather than premising its evaluation on the challenged product's reception in the market, the court in *C.R. Bard* found the defendant's intent dispositive.[68] Bard, the defendant, offered procompetitive reasons for its changes — namely that the innovations made its product easier to use — but the court found that "there was substantial evidence that Bard's real reasons for modifying the [product] were to raise the cost of entry" to potential competitors.[69] With the support of internal Bard documents indicating the changes brought no improvement to the product, the court held that the jury could reasonably conclude the modification "constituted 'restrictive or exclusionary conduct' in a market over which it had monopoly power."[70] Despite the focus on the defendant's intent instead of market acceptance, the *C.R. Bard* court parallels the *Berkey Photo* court in using the language of the rule of reason without employing its balance of competitive harms and benefits.

provided no justification for its product change or because the plaintiff had not rebutted the justification provided).

[62] *See Berkey Photo*, 603 F.2d at 288 ("Where a course of action is ambiguous, 'consideration of intent may play an important role in divining the actual nature and effect of the alleged anticompetitive conduct.'" (citing United States v. U.S. Gypsum Co., 438 U.S. 422, 436 n.13 (1978))).

[63] *See id.* at 287 ("The only question that can be answered is whether there is sufficient demand for a particular product to make its production worthwhile, and the response, so long as the free choice of consumers is preserved, can only be inferred from the reaction of the market.").

[64] *See* Kellie Lerner, *Tech-Tying vs. Section 2 of the Sherman Antitrust Act*, LAW360 (Dec. 4, 2014, 10:13 AM), http://www.rkmc.com/~/media/PDFs/Tech-Tying%20Vs%20%20Section%202%20Of%20The%20Sherman%20Antitrust%20Act.pdf [http://perma.cc/5EWC-Y47T] (describing the D.C. Circuit applying a balancing test, the Ninth Circuit presuming the legality of any tie that offers an improvement, the Federal Circuit analyzing intent, and the Second Circuit examining consumer demand).

[65] *See Berkey Photo*, 603 F.2d at 287–88.

[66] *See id.*

[67] *See* C.R. Bard, Inc. v. M3 Sys., Inc., 157 F.3d 1340, 1382 (Fed. Cir. 1998).

[68] *Id.*

[69] *Id.*

[70] *Id.*

Below, we examine three assumptions about the nature of physical products underlying the market acceptance and intent tests that do not apply to software products.

B. Assumed Characteristics in Markets of Physical Products

The limitations on the predatory innovation theory depend on three assumptions about the nature of markets in physical products: (1) lack of network effects; (2) concurrent existence of earlier versions of the product, which may be sold by the producer alongside newer versions or by consumers as used goods; and (3) meaningful and active user choice among products. We detail each of these below.

1. Weak Network Effects

Network effects are the positive externalities that accrue to a producer and consumers when multiple consumers use the product.[71] Hence, the more consumers using the product, the more useful it is to each individual. Products that capture network effects generally are those geared toward communication or interaction,[72] such as telephones,[73] email, and applications on operating systems.[74] The value accrued from each additional user goes directly to the user on the demand side; network effects thus differ from economies of scale, which reduce costs for suppliers.[75]

Strong network effects can create barriers to entry, helping a monopolist maintain its dominance in the market.[76] Yet courts developing the limitations on the theory of predatory liability generally ignored implicit arguments that network effects applied to the physical product markets at issue, or else discounted any network effects or first-mover advantages as being surmountable through reverse-engineering for compatibility.

In *Berkey Photo, Inc. v. Eastman Kodak Co.*, the Second Circuit rejected arguments that the necessity of a competitor's adherence to

[71] STUART MINOR BENJAMIN ET AL., TELECOMMUNICATIONS LAW AND POLICY 616 (Supp. 2002); Lemley & McGowan, *supra* note 7, at 488. Network effects are discussed in greater detail in the context of software product markets in Part IV.C.1.

[72] Lemley & McGowan, *supra* note 7, at 488.

[73] *Id.*

[74] *See* United States v. Microsoft Corp., 253 F.3d 34, 55 (D.C. Cir. 2001) ("That barrier — the 'applications barrier to entry' — stems from two characteristics of the software market: (1) most consumers prefer operating systems for which a large number of applications have already been written; and (2) most developers prefer to write for operating systems that already have a substantial consumer base."); *see also* Lemley & McGowan, *supra* note 7, at 491, 500 (discussing the role of "virtual network effects" in software, data sharing, and operating systems).

[75] *See* Speta, *supra* note 27, at 1279–80.

[76] *See id.* at 1279.

Kodak's standards created barriers to entry.[77] *Berkey Photo* had argued that Kodak "was in a position to set industry standards" in film formatting and that "[r]ivals could not compete effectively without offering products similar to Kodak's."[78] Kodak had maintained control of its film standards, thereby precluding other camera manufacturers — who could not obtain the information necessary to create cameras compatible with Kodak's film — from entering the market.[79] The court rejected the theory that the "chicken-and-egg" relationship between Kodak's film and its rivals' camera posed a barrier to entry, despite the fact that the success of Kodak's film made it a necessary component for competitors in the peripheral camera market.[80] Although competitors were essentially forced to adopt Kodak's physical standards, the court found the time it took competitors to reverse engineer new products to be part of the reward for Kodak's development of a new product.[81]

The Northern District of California similarly rejected an argument that the dominance of IBM's central processing units ("CPUs") created a duty for it to disclose information to allow competitors to create peripheral computer units.[82] Despite the fact that competitors depended on compatibility with IBM's CPUs, the court found that the ability to reverse engineer defeated any barrier to entry.[83] Although the courts did not explicitly frame their analyses in terms of network effects in *Berkey* and *IBM*, the holdings implicate a key commonality: users' dependence on a dominant product increased the demand for compatible peripheral products. In *Berkey Photo* and *IBM*, the plaintiffs contended that the number of users of the central product was so high that competition in peripheral markets could not take place unless competitors were able to make their products compatible with the dominant central product.[84] However, absent true network effects — the indirect network effect of demand being bolstered by complementary products[85] — the barrier to entry of mere popularity could not

[77] Berkey Photo v. Eastman Kodak Co., 603 F.2d 263, 285–87 (2d Cir. 1979).

[78] *Id.* at 279.

[79] *Id.*

[80] The court labels the film and camera together a "system." *Id.* at 285–86.

[81] *See id.* at 283, 285.

[82] *See* ILC Peripherals Leasing Corp. v. Int'l Bus. Machs. Corp., 458 F. Supp. 423, 428, 436 (N.D. Cal. 1978).

[83] *Id.* at 437.

[84] This argument implies that Kodak and IBM had become *de facto* standard setters in the market, but refused to make their standards open for manufacturers of peripheral products. *See Int'l Bus. Machs. Corp.*, 458 F. Supp. at 436; *Berkey Photo*, 603 F.2d at 285–87. Lemley and McGowan make a similar observation of Microsoft's position in its antitrust litigation, siding with the *Berkey Photo* and *IBM* courts that the role as *de facto* standard setter ought not necessarily evoke antitrust scrutiny. Lemley & McGowan, *supra* note 7, at 506.

[85] For an explanation of such indirect network effects, see *infra* note 146 and accompanying text.

sustain the challenge. Ultimately, the courts rejected the claims based on the plaintiffs' ability to catch up through reverse engineering despite not being "at the starting line when the whistle blew."[86]

2. Continued Availability of Older Products

Second, courts have relied on the fact that producers of physical products cannot completely eliminate the old product from the market. In *Berkey Photo*, the Second Circuit noted that Kodak had continued to sell its older products beside the newer, allegedly predatory product.[87] Because Kodak did not remove its older Kodacolor X film, consumers were "not compelled" to purchase either the new film or the new camera introduced beside it.[88] The court noted that the availability of older products — and thus, consumer ability to choose — might be dispositive. It wrote:

> [T]he situation might be completely different if, upon the introduction of the 110 system, Kodak had ceased producing film in the 126 size, thereby compelling camera purchasers to buy a Kodak 110 camera . . . . In such a case the technological desirability of the product change might bear on the question of monopolistic intent.[89]

Similarly, in deciding that Abbott Laboratories' continued modification of its branded TriCor drug constituted anticompetitive conduct, the District of Delaware considered the availability of older versions of the product.[90] The Hatch-Waxman Act[91] requires producers attempting to introduce generic drugs into the market to establish that the generic is "bioequivalent" to the existing branded drug already approved by the FDA.[92] The plaintiff alleged that Abbott's continued variation of TriCor, including a switch from capsules to tablets, had no justified basis and served only to preclude competitors from establishing bioequivalence and entering the market.[93] The court carefully followed the Second Circuit's logic in *Berkey Photo*, noting that "[c]onsumers who are free to choose among various products enjoy

[86] *Berkey Photo*, 603 F.2d at 285.

[87] *Id.* at 278 ("Meanwhile, by 1973 the 110 had taken over most of the amateur market from the 126, and three years later it accounted for nearly four-fifths of all sales.").

[88] *Id.* at 287.

[89] *Id.* at 287 n.39.

[90] *See* Abbott Labs. v. Teva Pharm. USA, Inc., 432 F. Supp. 2d 408, 421–22 (D. Del. 2006).

[91] Pub. L. No. 98-417, 98 Stat. 1585 (1984) (codified in scattered sections of 21 and 35 U.S.C. (2012)).

[92] 21 U.S.C. § 355(j)(2)(A); *Abbott*, 432 F. Supp. 2d at 414.

[93] *Abbott*, 432 F. Supp. 2d at 416.

the presence of competition rather than its absence."[94] Unlike Kodak in *Berkey Photo*, Abbott had pulled the prior iterations of TriCor from the market, precluding not only competitors from entering under Hatch-Waxman,[95] but also consumers from choosing among the versions of the medication. Without the continued sale of older versions of the product, there was no basis for judicial deference to Abbott's defense of innovation.[96] In both *Berkey Photo* and *Abbott*, the deciding factor was the availability of the older product; where the older product had been eliminated, there was no consumer choice and no defense of innovation.

3. Users Recognize Change

Finally, and relatedly, courts have assumed that the market recognizes the introduction of a new product and is able to accept or reject it. In *Berkey Photo*, the market's knowledge and acceptance of the new product were clear from Kodak's extensive advertising and the relative success of the new film.[97] Consumers had the choice to purchase either the new or old film; furthermore, advertising informed them of the new product's features. Although the advertising "bathe[d] [Kodak's] cause in the best light possible," it did not rise to a barrier to entry or curtail consumers' freedom of choice.[98] The *Abbott* court similarly noted that where consumers are free to make an informed choice among products, the court must defer to the market.[99] As a respected commentator has noted, where consumers are informed and able to choose products in an open market, courts have "no choice but to accept consumer sovereignty."[100]

Thus, the courts' deference to producers for product changes rests on the three assumptions described above. These assumptions not only follow from the case law but also make sense logically. If courts are going to rely on the market's reaction as an indicator of genuine innovation, it is necessary that consumers be free to make an informed choice uncompelled by network effects, be allowed to return to the old product if they find the new iteration unsatisfactory, and be aware that a change has occurred. As we demonstrate in the next Part, these assumptions do not hold for software markets, especially those on the Internet.

[94] *Id.* at 421.

[95] Removing the older product from the market and changing its status to "obsolete" with the FDA which precluded pharmacies from filling prescriptions for bioequivalent generics. *Id.* at 418.

[96] *Id.*

[97] Berkey Photo, Inc. v. Eastman Kodak Co., 603 F.2d 263, 287 (2d Cir. 1979).

[98] *Id.* at 287.

[99] *Abbott*, 432 F. Supp. 2d at 421.

[100] IIIA PHILLIP E. AREEDA & HERBERT HOVENKAMP, ANTITRUST LAW § 781 (4th ed. 2014).

IV. SOFTWARE PRODUCT REDESIGN AND MARKETS

Compared to their physical counterparts, software products are more likely to depend on network effects, enable the elimination of older iterations of the product, and support drastic changes that are imperceptible to the consumer. These characteristics are especially true for online web applications.

The success of software products strongly depends on network effects, and thus on interoperability — the degree to which the products are able to interact with other computer systems or software in an increasingly interconnected computing landscape. The standards-based nature of the Internet, along with SSOs in other contexts, has encouraged effective interoperability for software products, often increasing value and number of users and developers for those products.

In the digital world, competitors find it difficult to overcome a company's efforts to capture the benefits of network effects. In both the physical and digital landscapes, intellectual property rights can be used to exclude competitors from interoperating with a firm's product, if doing so would tread on those rights. Physical products may be designed around a patent to avoid infringement while still producing a competitive alternative; software products, however, may be technically prevented from accessing and using other components on which they depend.

Competition law was developed in the context of physical product markets and has not adapted to this distinction between physical and software products. Product redesign that breaks interoperability poses a danger to competition in markets of software products that is not addressed by the current predatory innovation theory.

A. What Are Software Products?

This Note is concerned with software products, products which are composed exclusively of code, or machine-interpretable instructions. This Section aims to provide a technical background and consistent terms to frame analysis of the industry. Computer science is a relatively young field, and terminology and definitions have changed substantially as technologies evolve. Early in the Information Age, computers were machines that read programs stored on punched cards. In order to redesign the programs, programmers had to reorganize cards and punch new ones.[101] Today, even the definition of computer depends on the context and may refer specifically to a personal computer or broadly to any device that can be programmed to carry out a

[101] *See generally* Dale Fisk, *Programming with Punched Cards*, COLUM. U. (2005), http://www.columbia.edu/cu/computinghistory/fisk.pdf [http://perma.cc/HQ5Q-ZU8F] (describing programming in the punched card era).

set of instructions.[102] We adopt the latter definition, including in its scope personal computers (e.g., smartphones, laptops, in-car computers, gaming consoles), servers, and mainframes.

1. Software

Software may be conceptualized as any set of machine-interpretable instructions that direct a computer to perform a task, as opposed to the physical components of a computer. We adopt this definition, although it is worth noting that several exist, even within the industry.[103] For the purposes of this analysis, software encompasses any non-physical components of a computer system, including system software, which manages a computer (e.g., operating systems, firmware) and application software, which cannot run on itself but depends instead on system software (e.g., social media platforms, web browsers, mobile apps, word processors, video games).

2. Interoperability

Most software interacts with other software, relying on interoperability: the ability to (1) exchange information and (2) use the exchanged information.[104] Information exchange depends on syntactic

[102] Charles Babbage's difference engine was an entirely mechanical computer capable of tabulating polynomial functions. *See* YOSHIHIDE IGARASHI, ET AL., COMPUTING: A HISTORICAL AND TECHNICAL PERSPECTIVE 91–92 (2014). Babbage also proposed an analytical engine "composed of a *store* (now called the computer memory) and *mill* (now called the central processing unit or CPU)." *Id.* at 92.

[103] *Compare Software*, OXFORD ENGLISH DICTIONARY, http://www.oed.com/view/Entry/183938 [http://perma.cc/BYT4-EQ55] ("programs and procedures required to enable a computer to perform a specific task, as opposed to the physical components of the system"), *with Welcome to SEVOCAB: Software and Systems Engineering Vocabulary*, INST. OF ELEC. & ELECS. ENG'RS (IEEE) COMPUTER SOC'Y, http://pascal.computer.org/sev_display/index.action [http://perma.cc/V63X-5KTQ] (search "software" in the term box for definitions from international standards) (providing a number of definitions for "software"). Sometimes a distinction is drawn between software and firmware, the latter being the lower-level, often read-only code on the boundary of hardware and software. *See* Int'l Org. for Standardization/Int'l Electrotechnical Comm'n, ISO/IEC 12207: Systems and Software Engineering — Software Life Cycle Processes, § 4.14 (2008), https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en [http://perma.cc/6HQD-N7QU] (specifying in the overview that "[s]oftware includes the software portion of firmware," but also defining firmware as the "combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device"). Distributed systems may be distinguished as software operating on networked computers, rather than on a single computer. *See generally* GEORGE COULOURIS ET AL., DISTRIBUTED SYSTEMS: CONCEPTS AND DESIGN (5th ed. 2011).

[104] *See generally* HERBERT KUBICEK, RALF CIMANDER & HANS JOCHEN SCHOLL, ORGANIZATIONAL INTEROPERABILITY IN E-GOVERNMENT: LESSONS FROM 77 EUROPEAN GOOD-PRACTICE CASES 85–96 (14th ed. 2011) (adopting cumulative layers of interoperability: technical, syntactic, semantic, and business process); *see also* Amit P. Sheth, *Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantic*, *in* INTEROPERATING GEOGRAPHIC INFORMATION SYSTEMS 5–29 (Michael Good-

interoperability, which allows systems to communicate and exchange data through packaging and transmissions mechanisms, such as common data formatting and protocol for data structure.[105] This information exchange takes place through an interface, which provides rules for possible interactions with the corresponding software implementation.[106] Semantic interoperability ensures that software can interpret the information for use.

An application programming interface ("API"), for example, is "a set of features and rules" for software components to interact with the software providing the API.[107] In web development, "API commonly means a set of standard methods, properties, events, and URLs" for interacting with a web platform.[108] System software developers document the interface specifications, which identify and explain these features and rules, and regularly publish them for others to use, sometimes even providing a software development kit that exposes those interactions as constructs in a programming language.[109] However, documentation is notoriously neglected and often out of date or incomplete, meaning the specifications that set forth purportedly permissible interactions may be incorrect, while other technically possible interactions could be undocumented.[110] As a result, some do not believe detailed documentation to be useful or even worth the cost and time of development.[111] The API provider may also limit access, disclosure, and documentation to interfaces to the extent deemed necessary or desirable for interoperability. If limited public or official information is available, an application developer may rely on undocumented behaviors or attempt to reverse engineer the interface specification.[112] Naturally, open interface specifications enable greater

child et al. eds., 1999) (discussing system and structure interoperability); MATHIAS USLAR ET AL., THE COMMON INFORMATION MODEL CIM (2012).

[105] *See* KUBICEK ET AL., *supra* note 104, at 87.

[106] *See generally* GARY STRINGHAM, HARDWARE/FIRMWARE INTERFACE DESIGN: BEST PRACTICES FOR IMPROVING EMBEDDED SYSTEMS DEVELOPMENT 2 (2010) (describing the hardware/firmware interface).

[107] *API*, MOZILLA DEVELOPER NETWORK, https://developer.mozilla.org/en-US/docs/Glossary/API [http://perma.cc/E29B-Q4G9].

[108] *Id.*

[109] *See, e.g.*, *Microsoft API and Reference Catalog*, MICROSOFT DEVELOPER NETWORK, https://msdn.microsoft.com/en-us/library/ [http://perma.cc/MZ2D-UYGA]; *APIs Explorer*, GOOGLE, https://developers.google.com/apis-explorer/ [http://perma.cc/5M2R-SLP5].

[110] *See, e.g.*, IAN SOMMERVILLE, SOFTWARE ENGINEERING 64 (9th ed. 2011) (noting that documentation is "not of much use for long-term system maintenance"). Documentation is often low priority, so emergency fixes may be made and forgotten, leaving documentation and code unaligned. *See id.* at 239. For a discussion of types of documentation, see generally Ian Sommerville, SOFTWARE ENGINEERING, *Documentation*, http://www.softwareengineering-9.com/Web/ExtraChaps/Documentation.pdf [http://perma.cc/R564-T7GC].

[111] *See* SOMMERVILLE, SOFTWARE ENGINEERING, *supra* note 110, at 150, 155.

[112] *See* JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL: INTELLECTUAL PROPERTY AND INTEROPERABILITY IN THE GLOBAL SOFTWARE INDUSTRY 14 (1995).

interoperability, while closed interface specifications obstruct it. As part of publishing APIs, a firm may also make support statements indicating under what circumstances and for how long that API will be supported by the system software — i.e., how long the system software will interact with application software according to those interface specifications.[113]

3. Dependencies

Where software depends on the behavior of hardware or other software to perform a task, this relation is called a dependency.[114] More than one dependency may exist for a given task, and the set of dependencies may go in either direction — requiring specific behavior from only one component if unidirectional or from both interoperating components if bidirectional. For two components to be interoperable, all dependencies must be satisfied. There may also be optional dependencies, which enable additional features if satisfied but do not impede proper functioning of the program if violated.[115]

Software designed to satisfy a dependency set forth by another hardware or software component may not do so for various reasons, such as: the software may have been implemented incorrectly; the dependency may have been documented incorrectly or inadequately, such that the software met the documented requirements set forth in the specification but failed to satisfy the dependency; the software may have a dependency of its own that creates a circular dependency;[116] or, very rarely, single-event upsets, or soft errors, may be caused by cosmic rays or alpha particle emissions.[117]

[113] *See infra* notes 120–26 and accompanying text.

[114] The word "dependency" may also be used to describe the depended-upon behavior or the information describing the depended-upon behavior. *See, e.g.*, James Weir, *"I'm My Own Grandpa" — Avoiding Dependency Hell*, USHARESOFT (Oct. 15, 2013), https://blog.usharesoft.com/index.php?article44/i-m-my-own-grandpa-avoiding-dependency-hell [http://perma.cc/8WYH-5EQR] ("A dependency is a piece of information in a software package that describes which other packages it requires to function correctly."). For an explanation of the types of data or instructions that may be involved in the dependency, see generally NORMAN WILDE, UNDERSTANDING PROGRAM DEPENDENCIES 4–5 (Aug. 1990), https://www.sei.cmu.edu/reports/90cm026.pdf [http://perma.cc/E6B2-CEHC]; Marcelo Cataldo et al., *Software Dependencies, Work Dependencies, and Their Impact on Failures*, 35 IEEE TRANSACTIONS ON SOFTWARE ENG'G 864 (2009), http://www.herbsleb.org/web-pubs/pdfs/cataldo-software-2009.pdf [http://perma.cc/2DZHMFPA].

[115] Meriem Belguidoum & Fabien Dagnat, *Dependency Management in Software Component Deployment*, 182 ELECTRONIC NOTES IN THEORETICAL COMPUTER SCI. 17, 20 (2007), *available at* http://www.sciencedirect.com/science/article/pii/S1571066107003830 [http://perma.cc/R7ZJ-UG4J].

[116] *See* WILDE, *supra* note 114; Marcelo Cataldo et al., *supra* note 114.

[117] *See* Santosh Kumar et al., *Soft Error Issue and Importance of Low Alpha Solders for Microelectronics Packaging*, 34 REVS. ON ADVANCED MATERIALS SCI. 185, 186–88 (2013), http://www.ipme.ru/e-journals/RAMS/no_23413/07_23413_kumar.pdf [http://perma.cc/W4RJ-KF6M]; ACTEL, UNDERSTANDING SOFT AND FIRM ERRORS IN

If a dependency is not satisfied, the software may perform a task incorrectly or fail to perform the task altogether. For example, application software may have dependencies that must be satisfied before the software can be installed and available to run on system software. Such a dependency may be violated where minimum system requirements are not met, a required local software library is unavailable, or the application software was not properly installed, was removed, or is temporarily inaccessible (e.g., because of security permissions). Software may also depend on other software running on a remote system and being accessible via the Internet or another network. Such a dependency may be violated where the remote software does not behave consistently with the dependency, perhaps due to a decision to no longer support that type of dependency (e.g., when a new version of an API is released), unintentional errors in implementation, or lack of network connection.[118] The provider of the remote software controls the continued availability of the dependency. Some of the more frustrating dependency-related issues gave rise to the term "dependency hell."[119]

B. Software Product Redesign and Breaking Interoperability

Implementing changes in software products may be faster and less conspicuous than in physical products. For software products, changes may be distributed in the form of automatic software updates, prompts to update when using the product, or the release of a new version. Other products, in particular for web-based platforms and services, may simply change the user or application interfaces; because they are hosted remotely and accessed through the web browser or another application, the interfacing user or application is forced to use the updated version of the web-based technology.[120] Because distribution to users is relatively painless and unlikely to require special

SEMICONDUCTOR DEVICES 1 (Dec. 2002), http://www.microsemi.com/document-portal/doc_download/130765-understanding-soft-and-firm-errors-in-semiconductor-devices-questions-and-answers [http://perma.cc/X5VB-P8VB].

[118] *Cf.* MICHAEL JANG, LINUX ANNOYANCES FOR GEEKS: GETTING THE MOST FLEXIBLE SYSTEM IN THE WORLD JUST THE WAY YOU WANT IT 325–33 (2006) (describing how to resolve dependency issues when a software package has dependencies on specific versions of other software packages).

[119] Problems could include having many dependencies, long chains of dependencies, conflicting dependencies, and circular dependencies. *See id.* at 325; Weir, *supra* note 114; *cf.* Randall Munroe, *Dependencies*, XKCD (June 16, 2010), http://xkcd.com/754/ [http://perma.cc/RL2S-G94Y] (comic about circular dependencies).

[120] *See, e.g.*, Russ Garrett, *The Future of Last.fm Radio APIs*, LAST.FM (Mar. 24, 2009, 5:54 PM), http://www.last.fm/group/Last.fm+Web+Services/forum/21604/_/517212 [http://perma.cc/U5L7-9JGL] (describing Last.fm's development of a public, documented API to replace its old, undocumented API that allowed streaming); *An Updated News Feed*, FACEBOOK (Mar. 6, 2014), https://www.facebook.com/business/news/An-Updated-News-Feed [http://perma.cc/DGR3-S7GV] ("These changes will roll out in the coming weeks. As a marketer on Facebook, you don't need to take any action.").

marketing, software products may have more rapid and frequent, smaller-scale redesigns.

Firms may be transparent about how redesigns will affect interoperability, with support statements that specify a duration for support[121] or that generally encourage development of interoperable, complementary products.[122] To facilitate a transition, services may support multiple versions of an API and provide guidance about how long old versions will be supported after a newer version has been released. For example, Facebook typically supports an API version for one year following the introduction of a new API version.[123] Thus, Facebook stopped supporting its v1.0 API on April 30, 2015, one year after its v2.0 API was introduced.[124] The new API made several changes, including removing the ability to look at the data of users' friends and implementing a new login system.[125] While some applications were unaffected, others shut down due to the lost functionality.[126]

Support statements are important because, relative to physical products, software products can be redesigned to more reliably and completely preclude interoperability. Physical products may be inspected, taken apart, and reverse engineered. While there are also methods to reverse engineer information about software — by the developer or developers of complementary products, in order to compensate for poor documentation or nondisclosure — they are limited.[127]

A firm can make its product a black box. Using software anti-tamper technology, a firm can deliberately obfuscate code to make it difficult to understand or use an obfuscator to transform the code with

[121] *See, e.g.*, *Facebook Platform Changelog*, FACEBOOK, https://developers.facebook.com/docs/apps/changelog [http://perma.cc/LTG4-RMMG].

[122] *See Setting the Tone for Your API Terms of Service*, API EVANGELIST (Mar. 21, 2011), http://apievangelist.com/2011/03/21/setting-the-tone-for-your-api-terms-of-service/ [http://perma.cc/F898-FDUH] (providing excerpts of several API terms of service).

[123] *See Facebook Platform Versioning*, FACEBOOK, https://developers.facebook.com/docs/apps/versions [http://perma.cc/FR4V-S6YM]; *Facebook Platform Changelog*, FACEBOOK, *supra* note 121 (noting that each version of Facebook's API will be supported for at least two years from release).

[124] *Facebook Platform Changelog*, FACEBOOK, *supra* note 121 (noting that the v1.0 API was "[u]navailable as of April 30, 2015").

[125] Josh Constine, *Facebook Is Shutting Down Its API for Giving Your Friends' Data to Apps*, TECHCRUNCH (Apr. 28, 2015), http://techcrunch.com/2015/04/28/facebook-api-shut-down/ [http://perma.cc/959R-S9N6].

[126] *Id.*

[127] *See generally* Elliot J. Chikofsky & James H. Cross II, *Reverse Engineering and Design Recovery: A Taxonomy*, 7 IEEE SOFTWARE 13 (1990) (describing uses and a taxonomy for reverse engineering of software); Conference, Yijun Yu et al., *RETR: Reverse Engineering to Requirements*, 12 WORKING CONF. ON REVERSE ENG'G (Nov. 7–11, 2005); *see also Coders' Rights Project Reverse Engineering FAQ*, ELECTRONIC FRONTIER FOUND., https://www.eff.org/issues/coders/reverse-engineering-faq [http://perma.cc/32KF-2Y7B] (discussing legal implications of reverse engineering of software).

various techniques.[128] Those firms that do make limited information available may also prohibit reverse engineering under the terms of use. A closed source or limited API, for example, may allow a producer to maintain a commercial advantage over third-party software.[129] Finally, software products may be technically prevented from accessing and using other components on which they depend. If a firm controls use of and access to interfaces facilitating interoperability, it may sever that use or access by reference to a specific product or by restricting or barring all competing product interactions.[130]

In addition to an explicit denial of resources, a change in functionality may lead to a break in interoperability. For example, when Google dropped support of XMPP, an open messaging protocol, the Electronic Frontier Foundation criticized the break in interoperability with free and open source chat clients and servers.[131] Google, however, appeared to justify the change as being necessary due to the new features and technical demands of a "unified messaging platform."[132]

Software engineering strives to develop modular code that is well understood and documented. Still, a developer may write software that depends on "ill-understood, immature code," especially when the benefits, such as short-term development time and application running speed, outweigh the expected costs of future modification and debugging.[133] Similarly, when firms choose to develop products that depend

[128] *See generally* Avinash R. Desai et al., *Interlocking Obfuscation for Anti-Tamper Hardware*, PROC. OF THE EIGHTH ANN. CYBER SECURITY & INFO. INTELLIGENCE RES. WORKSHOP 1 (2013), http://www.faculty.ece.vt.edu/chaowang/pubDOC/DesaiHWNH13.pdf [http://perma.cc/4KBP-6348].

[129] *See Developer Agreement & Policy*, TWITTER, https://dev.twitter.com/overview/terms/agreement-and-policy [http://perma.cc/PV48-CVQV] ("You will not or attempt to (and will not allow others to) 1) reverse engineer, decompile, disassemble or translate the Twitter API, or otherwise attempt to derive source code, trade secrets or know-how in or underlying any Twitter API or any portion thereof . . . .").

[130] For example, a website may constrain third-party applications by limiting the number of users that may access the website through that application, *see supra* notes 16–19 and accompanying text, or by not providing access to specific features offered by the full website, *see* Isaach, Twitter Staff, Comment to *Search API and the New Complete Tweet Index Announcement*, TWITTER (Nov. 19, 2014, 1:01 PM), https://twittercommunity.com/t/search-api-the-new-complete-tweet-index-announcement/27883/3 [perma.cc/V3YN-AUXY] ("We don't have any plans to make the full historical index available for free via the public REST API.").

[131] *See* Parker Higgins, *Google Abandons Open Standards for Instant Messaging*, ELECTRONIC FRONTIER FOUND. (May 22, 2013), https://www.eff.org/deeplinks/2013/05/google-abandons-open-standards-instant-messaging [http://perma.cc/NT83-E5VW].

[132] *See* Ellis Hamburger & Dieter Bohn, *Exclusive: Inside Hangouts, Google's Big Fix for Its Messaging Mess*, VERGE (May 15, 2013, 1:58 PM), http://www.theverge.com/2013/5/15/4318830/inside-hangouts-googles-big-fix-for-its-messaging-mess [http://perma.cc/6ZEM-JRFP].

[133] Bruce D. Abramson & Dmitri L. Mehlhorn, *The Fettered Liberty To Integrate: Legal Implications of Software Engineering*, 10 B.U. J. SCI. & TECH. L. 209, 214–16 (2004) (noting, in the context of software integration, that "good software design develops 'modules'" that may be discarded or rewritten and recognizing that "integration of ill-understood,

on a third-party platform or other code, they choose what those dependencies are. If there is no contractual obligation for the third party to maintain support for, or even availability of, that platform or other code, the firm can anticipate the risk that the software will later be closed to outside parties.

It is worth noting that product redesign is possible even for firmware, which is traditionally thought of as being more tightly coupled to the hardware for which it is produced, and thus as having less flexibility to change because the underlying hardware must be able to support any features. Typically, the hardware vendor produces the firmware and the operating system vendor writes a device driver that provides an interface for other software to communicate with the hardware.[134] Firmware is sometimes discussed as if it were read-only,[135] but it can be updated to fix bugs or add features.[136] For example, a recent firmware update for Nikon cameras enables continuous shooting and unlimited exposure time — functionalities that were physically possible before the update, but could not be accessed by software.[137] Similarly, a firmware update could break interoperability or provide a competitive advantage; a solid-state drive ("SSD") producer could revise firmware and introduce a new instruction to improve performance, but only disclose or allow access to select software producers.

Given the many variables, it may be difficult to characterize how or why a product redesign breaks interoperability. Non-interoperability may result from changes to dependencies documented in support statements, or to behavior that is undocumented but still relied upon. These changes may explicitly restrict access to resources or otherwise change a behavior (e.g., to provide new functionality) in a way that no longer satisfies dependencies. The developer may not be aware that the change will result in non-interoperability, or explicitly decide to no longer support interoperability (e.g., because the party providing it decides the cost to continue providing it is too high). A break in interoperability may also be unintentional (e.g., due to a disaster impacting the physical location at which the dependency was running and a failure on the part of the provider to have made a back-up).

immature code fragments likely to require further modification and debugging is invariably a bad idea").

[134] *See generally* STRINGHAM, *supra* note 106, at 1 (describing firmware and device drivers).

[135] *See supra* note 104 and accompanying text.

[136] *See* Michael Zhang, *Nikon D4s Gets Unlimited Burst and Exposure Time via Firmware Update*, PETAPIXEL (Apr. 22, 2015), http://petapixel.com/2015/04/22/nikon-d4s-gets-unlimited-burst-and-exposure-time-via-firmware-update/ [http://perma.cc/3VT2-NRLK].

[137] *See id.*

C. Differences in Software Product Markets

Software product markets[138] do not reflect the three characteristics underlying antitrust analyses in physical product markets. First, software product markets are more likely to exhibit strong network effects, which dominant firms may use for competitive advantage. This is particularly true due to the difficulties in reverse engineering and the ability to deny access to resources. Second, there is effectively no market for used software products because users often purchase a software license but not ownership, and product updates often overwrite or modify an existing product. Third, a developer may distribute an update for a software product with minimal impact on user experience, so users may not be aware of the nature of changes.

1. Strong Network Effects

Software product markets often exhibit and depend on network effects, the positive externalities that accrue as use increases, discussed above.[139] These network effects are not only stronger in many software product markets, but they also have a greater impact. Once a firm has achieved dominance by benefitting from complementary third-party products, it can completely restrict interoperability with competitors. In *United States v. Microsoft Corp.*, for example, the court found that developers will only create applications for an operating system if other developers have already done so to attract consumers.[140] This "chicken-and-egg" relationship between applications for an operating system and its number of users created a barrier to entry that allowed the court to infer market power.[141]

Network effects are stronger if connection to others in the network is inherently valuable;[142] a user may derive inherent value from using the product as well as network value from others' use of the product. Products designed for users to interact with each other, such

[138] For a discussion of the relevant markets for software products, see generally Andrew Chin, *Antitrust Analysis in Software Product Markets: A First Principles Approach*, 18 HARV. J.L. & TECH. 1 (2004); Jared Kagan, *Bricks, Mortar, and Google: Defining the Relevant Antitrust Market for Internet-Based Companies*, 55 N.Y.L. SCH. L. REV. 271 (2010); Spencer Weber Waller, *Antitrust and Social Networking*, 90 N.C. L. REV. 1771, 1776–84 (2012) (characterizing social networking websites as competing in product, geographic, and data markets).

[139] *See supra* notes 71–76 and accompanying text.

[140] *See* 253 F.3d 34, 55 (D.C. Cir. 2001) ("That barrier — the 'applications barrier to entry' — stems from two characteristics of the software market: (1) most consumers prefer operating systems for which a large number of applications have already been written; and (2) most developers prefer to write for operating systems that already have a substantial consumer base.").

[141] *Id.*

[142] *See* William H. Page & Seldon J. Childers, *Antitrust, Innovation, and Product Design in Platform Markets: Microsoft and Intel*, 78 ANTITRUST L.J. 363, 369 (2012).

as social networks and chat clients, require widespread adoption among users to maximize both inherent and network value: a user can derive value from these products only because other users exist, and a larger user base increases that inherent value while also offering value by virtue of their use (e.g., through contributions and posts on social networking websites). Thus, young companies often open the APIs of their products to competitors to allow interoperability and extend their reach and adoption.[143] When a dominant firm subsequently breaks interoperability, that break can have a decisive impact on smaller competitors.

Different types of network effects exist, among them (1) direct, in which increased use increases the value of the product itself;[144] (2) indirect, in which increased use leads to development of complementary products, such as applications for a specific platform, which in turn increases the value of the product;[145] and (3) two-sided, in which increased use by one set of users increases the value of a complementary product and vice-versa.[146] In other words, the structure of the network influences who benefits from whom.[147]

Network effects can act like a ratchet, allowing firms to become increasingly dominant: the more valuable it is for consumers to share a network implicated by the product, the more complementary products that will be created for it and the greater their value, which in turn increases the product's value. This cycle of network effects can create substantial barriers to entry and exit, wherein consumers may be reluctant to use software that is not already popular or widely used, despite comparable or even superior functionality.[148]

Market power may also be obtained and maintained via lock-in, namely specific configurations of hardware and/or operating systems and platform-specific software that make it costly to switch platforms

[143] *Cf.* Ben Popper, *Twitter Follows Facebook Down the Walled Garden Path*, VERGE (July 9, 2012, 10:42 AM), http://www.theverge.com/2012/7/9/3135406/twitter-api-open-closed-facebook-walled-garden [http://perma.cc/D467-4GYA] (commenting on Twitter's move from an open platform).

[144] *See generally* Michael L. Katz & Carl Shapiro, *Network Externalities, Competition, and Compatibility*, 75 AM. ECON. REV. 424 (1985).

[145] *See generally* Nicholas Economides & Steven C. Salop, *Competition and Integration Among Complements, and Network Market Structure*, 40 J. INDUS. ECON. 105 (1992) (analyzing competition and integration among complementary products).

[146] *See generally* Geoffrey G. Parker & Marshall W. Van Alstyne, *Two-Sided Network Effects: A Theory of Information Product Design*, 51 MGMT. SCI. 1494 (2005) (explaining two-sided network externalities and how they enable profitable provision of products to one set of users for free).

[147] *See, e.g.*, Zsolt Katona, Peter Pal Zubcsek & Miklos Sarvary, *Network Effects and Personal Influences: The Diffusion of an Online Social Network*, 48 J. MARKETING RESEARCH 425 (2011) (discussing diffusion in online social networks in part as a function of the local network structure and user characteristics).

[148] *See* Waller, *supra* note 138, at 1786–90 (discussing network effects and stickiness for social networking websites).

despite some increased price.[149] The Electronic Frontier Foundation and the Free Software Foundation have objected to the use of DRM technologies as a similarly anticompetitive practice, designed to lock consumers into a platform, device, or other technology by preventing export to another.[150] One argument is that its use can create a "horizontal lock-in" that encourages or requires the consumer to continue using a given technology.[151] Not only is the consumer locked in for that specific content, but he or she also grows familiar and comfortable with the platform, device, software, and compatible products — another benefit to the producer.[152] However, they argue that this restriction lowers consumer valuation without distributing any benefit to the copyright holder.[153]

Network effects, combined with limitations on interoperability, can mirror the foreclosure of consumer choice and exclusionary effects of traditional tying arrangements[154] and unilateral refusals to

[149] *Id.* at 1791–92 (discussing market power via lock-in for social networking websites).

[150] Fred von Lohmann, an intellectual property attorney, has authored numerous articles against use of DRM. *See, e.g.*, Fred von Lohmann, *Fair Use and Digital Rights Management: Preliminary Thoughts on the (Irreconcilable?) Tension Between Them*, ELECTRONIC FRONTIER FOUND. (Apr. 16, 2002), https://w2.eff.org/IP/DRM/cfp_fair_use_and_drm.pdf [http://perma.cc/635E-T9HF]; Fred von Lohmann, *FairPlay: Another Anticompetitive Use of DRM*, ELECTRONIC FRONTIER FOUND. (May 25, 2004), https://www.eff.org/deeplinks/2004/05/fairplay-another-anticompetitive-use-drm [http://perma.cc/8J9S-AMRB]. The Free Software Foundation has led campaigns against specific uses of DRM. Letter from Richard Stallman, President of the Free Software Found., to the Bos. Pub. Library (Jan. 30, 2006), https://www.fsf.org/campaigns/bpl.html [http://perma.cc/U7EN-R27H] (urging the Boston Public Library to "terminate its association with OverDrive Audio Books, and adopt a policy of refusing to be agents for the propagation of Digital Restrictions Management"); Matt Lee, *Join the FSF in Calling on Libraries To Eliminate DRM*, FREE SOFTWARE FOUND. (Feb. 7, 2008, 4:20 PM), https://www.fsf.org/news/DRMLibraryAction.html [http://perma.cc/8MG8-SPQ5].

[151] *See* Rene Ritchie, *Horizontal vs. Vertical Lock-In: Until DRM Dies, iTunes Is No Worse than Google or Amazon*, IMORE (Jan. 22, 2014, 9:11 PM), http://www.imore.com/until-drm-dies-itunes-no-worse-lock-in-amazin-google-anyone-else [http://perma.cc/J78B-6VNX].

[152] *See The Customer Is Always Wrong: A User's Guide to DRM in Online Music*, ELECTRONIC FRONTIER FOUND., https://www.eff.org/pages/customer-always-wrong-users-guide-drm-online-music [http://perma.cc/Q9L3-EQSE]. Amazon was able to achieve a dominant position in the ebook market by building DRM into their ebook formats. Mike Masnick, *How Publishers Repeated the Same Mistake As Record Labels: DRM Obsession Gave Amazon Dominant Position*, TECHDIRT (Feb. 13, 2012, 9:54 AM), https://www.techdirt.com/articles/20120210/01364817725/how-publishers-repeated-same-mistake-as-record-labels-drm-obsession-gave-amazon-dominant-position.shtml [http://perma.cc/8PNC-HFN4]; *see also* Mike Masnick, *Why Are Book Publishers Making the Same Mistake the Record Labels Made with Apple?*, TECHDIRT (Feb. 27, 2009, 8:44 AM), http://www.techdirt.com/articles/20090227/0128303920.shtml [http://perma.cc/6RFH-GG7K]; Farhad Manjoo, *Fear the Kindle: Amazon's Amazing E-book Reader Is Bad News for the Publishing Industry*, SLATE (Feb. 26, 2009, 5:14 PM), http://www.slate.com/articles/technology/technology/2009/02/fear_the_kindle.html#p2 [http://perma.cc/DJ8Y-NZUE].

[153] *See* von Lohmann, *FairPlay*, *supra* note 150.

[154] *See infra* Part V.A.

deal.[155] This is because such arrangements further incentivize consumers to remain with the monopolist, as emerging competitors do not have access to the same content or share the same widespread use. Physical products are less likely to depend on network effects in order to be operable and consequently, refusals to deal are less likely to deny access to essential content or other resources. Further, some software changes can absolutely deny a competitor access to content or products that cannot be remedied by reverse engineering. The harm of network effects in software markets, then, is more than a mere head start or first-mover advantage for the monopolist.

Robert Cass describes the unique nature of software product markets and the implications for antitrust enforcement based on conduct that may not be readily distinguishable from permissible competitive conduct:

> In rapidly changing high-technology industries, the problems can be especially acute and can threaten innovation as well as competition. High-technology industries are often characterized by large up-front investments in research and development, intense competition for breakthrough innovations, large economies of scale, and potential "network effects" that produce big gains over some time period for the most successful innovators. These are the characteristics of "winner-take-all" or "winner-take-most" markets.[156]

Despite the implications of network effects, motivated software consumers can often switch from one provider or technology to another at minimal or zero cost, and may opt to use more than one that serve similar functions.[157] Social networking websites and computer applications (e.g., web browsers or games available in app stores) are often freely available, as they derive revenue from advertisements and other user-dependent streams.[158] This may counter in part some of the

[155] *See infra* Part V.B.

[156] Robert A. Cass, *Antitrust for High-Tech and Low: Regulation, Innovation, and Risk*, 9 J.L. ECON, & POL'Y 169, 174–75 (2013) [hereinafter Cass, *Regulation, Innovation, and Risk*]; *see also* Ronald A. Cass, *Antitrust and High-Tech: Regulatory Risks for Innovation and Competition*, 14 ENGAGE: J. FEDERALIST SOC'Y PRAC. GROUPS 25, 25 (2013).

[157] *See generally* Cass, *Regulation, Innovation, and Risk*, *supra* note 156, at 195–96 ("Despite the networks they have established, each of these businesses is also notable for the relative ease with which consumers can switch from one provider or technology to another . . . [or] add additional products or services from multiple providers at minimal or zero cost."). While some software products, such as early financial software, do not exhibit strong network effects, they may still be subject to the test we propose, which also examines whether older versions of the software remain on the market and whether consumers are aware of a change in interoperability when choosing among products.

[158] *See id.*

lock-in network effects, but only where competing products have emerged despite barriers to entry.

2. Elimination of Older Products by Licensing and Overwriting

As discussed above, the continued availability of older products — and thus, consumers' ability to choose — is crucial to courts' assessments of anticompetitive conduct by market acceptance.[159] Producers of software products can prevent a used good market from emerging by (1) providing users with a software license, rather than ownership; (2) controlling software use with DRM technologies; and (3) having product updates overwrite or modify an existing product, rather than providing concurrent support and licenses for an old and new version of the product.

Software is a durable good, arguably more durable than physical products because it wears out only due to technological change and planned obsolescence.[160] Firms have avoided the Coase Conjecture drive toward selling products at marginal cost in part by licensing (rather than selling) software, with terms that restrict the consumer's ability to transfer or resell the product.[161] Software products also use DRM technologies to control their use — such as installation, access, and copying content — after sale.[162] DRM can take many forms: Origin, the digital distribution platform for Electronic Arts, requires users to authenticate with an online server to limit the number of installations per twenty-four hours,[163] while Blizzard's Diablo III employs always-online DRM that requires an Internet connection to play, even when using local content during single-player mode.[164] Because

[159] *See supra* Parts III.B.2–3.

[160] *See* COMPETITION, INNOVATION AND THE MICROSOFT MONOPOLY: ANTITRUST IN THE DIGITAL MARKETPLACE 36 (Jeffrey A. Eisenach & Thomas M. Lenard eds., 1999). Other strategies for precluding a used software market include ensuring a flow of new customers and continuing to improve the product to create demand. *Id.*

[161] *See id.*; David Kravets, *Guess What, You Don't Own That Software You Bought*, WIRED (Sept. 10, 2010, 2:01 PM) http://www.wired.com/2010/09/first-sale-doctrine/ [http://perma.cc/4562-Z2L5]. Microsoft has even begun to move from licenses limiting transfer and sale to licenses that have limited duration, offering software-by-subscription plans for its office suite. *See Buy Office 365*, MICROSOFT, https://products.office.com/en-us/buy/office [http://perma.cc/47NZ-QSUK].

[162] *See* S.R. Subramanya & Byung K. Yi, *Digital Rights Management*, IEEE POTENTIALS, Mar.–Apr. 2006, at 31, 31 (noting that the term may also refer broadly to policies, tools, and other mechanisms for guiding use of any digital information, ranging from audio and video to text); *cf. Software Licensing and Protection Services*, MICROSOFT DEVELOPER NETWORK, https://msdn.microsoft.com/en-us/library/bb931699.aspx [http://perma.cc/EEN7-6KCF] (providing a suite of licensing components for developers).

[163] Emanuel Maiberg, *Changing PC Hardware Too Often Can Lock You Out of Origin Games, EA Confirms*, GAMESPOT (Mar. 28, 2015), http://www.gamespot.com/articles/changing-pc-hardware-too-often-can-lock-you-out-of/1100-6426240/ [http://perma.cc/6PW9-7SNL].

[164] Industry observers posit three reasons for Blizzard's choice to use always-online DRM: preventing cheating in an auction house, facilitating character online-offline transi-

the first sale doctrine may not apply to software licensees,[165] these constraints together prevent emergence of a robust used goods market for software products.

Some software products, such as websites and social networking platforms, are provided remotely and thus can be completely overwritten when updated; even if a backup of a prior version is saved, it is not hosted in the same location and may not be accessible to third-party developers.[166] Similarly, when many software products are updated, or when a new version is installed, only the newest version of the product continues to be available.[167] A firm may even prevent a user who updates or installs a new version of the product from reverting to or continuing to use an earlier version on the same machine.[168] Because the update merely patches or completely replaces the existing software, there is no old product to use in the alternative or resell.

tions, and preventing piracy. However, gamers widely criticized its use as it interfered with gameplay. Dave Their, *The Diablo 3 Launch and the Perils of Online-Only*, FORBES (May 15, 2012, 12:17 PM), http://www.forbes.com/sites/davidthier/2012/05/15/the-diablo-3-launch-and-the-perils-of-online-only/ [http://perma.cc/FRU6-BCKX]; *see also* Simon Jary, *Diablo III Players Angry as Hell at Launch Chaos*, PC ADVISOR (May 16, 2012), http://www.pcadvisor.co.uk/news/game/3358011/diablo-iii-players-angry-as-hell-at-launch-chaos/ [http://perma.cc/A6EJ-E3W2] (noting server difficulties coping with massive traffic, "Error 37," and additional problems caused by maintenance).

[165] *See generally* Terence Leong, *When the Software We Buy Is Not Actually Ours: An Analysis of* Vernor v. Autodesk *on the First Sale Doctrine and Essential Step Defense*, 10 NW. J. TECH. & INTELL. PROP. 239 (2012) (discussing the circuit split on whether and when a software licensee is considered an owner for purposes of 17 U.S.C. § 117 (2012)). *Compare* Vernor v. Autodesk, Inc., 621 F.3d 1102, 1111 (9th Cir. 2010) (holding that a user under a restrictive software licensing scheme is not an owner and therefore cannot invoke affirmative copyright infringement defenses, such as the first sale doctrine), *with* DSC Commc'n Corp. v. Pulse Commc'n, Inc., 170 F.3d 1354, 1360 (Fed. Cir. 1999) (holding that a licensee may be an owner for the purposes of 17 U.S.C. § 117 (2012), limiting exclusive rights for computer programs), *and* Krause v. Titleserv, Inc., 402 F.3d 119, 123–24 (2d Cir. 2005) (adopting the *DSC* standard).

[166] *See* Jill Lepore, *The Cobweb*, NEW YORKER (Jan. 26, 2015), http://www.newyorker.com/magazine/2015/01/26/cobweb [http://perma.cc/NA49-KT3J] ("Web pages don't have to be deliberately deleted to disappear. Sites hosted by corporations tend to die with their hosts. When MySpace, GeoCities, and Friendster were reconfigured or sold, millions of accounts vanished.").

[167] *Cf. Changing the Product Code*, MICROSOFT DEVELOPER NETWORK, https://msdn.microsoft.com/en-us/library/aa367850%28v=vs.85%29.aspx [https://perma.cc/N25Q-AWM3] (noting that the product code must be changed to allow "[c]oexisting installations of both original and updated products on the same system").

[168] *Cf. How To Downgrade from Windows 8*, MICROSOFT, https://support.microsoft.com/en-us/kb/2832566 [https://perma.cc/BC7F-4YZA] ("There are no downgrade rights for retail versions of Windows 8. If you upgraded Windows by using a retail version of Windows 8, you have to reinstall your earlier version of Windows by using the recovery or installation media that was included with your PC."); Eric Limer, *How To Go Back to iOS 6 (If You Can at All)*, GIZMODO (Sept. 19, 2013, 2:20 PM), http://gizmodo.com/how-to-go-back-to-ios-6-if-you-can-at-all-1326976091 [http://perma.cc/A796-REQB] ("At some point — soon — this process [to downgrade to iOS 6] is going to stop working. Maybe it already has. Once it does you're probably stuck. And probably forever.").

3. Changes Can Have Little User or Consumer Impact

While users may be capable of switching products in response to a redesign, they may not even be aware that any changes were made. A dominant firm may implement changes that have minimal or no impact on the user experience for its current users, but that restrict or completely break interoperability with a competing firm's product, leading users to turn to that dominant firm. Current users may not even realize that the change has been made — after redesign, the product update may be distributed by automatic updates for application and system software or simply by updating the web service with the changes.[169] For example, when Twitter changed its API to limit user tokens for third-party applications, the changes did not affect current users of those applications and of the official Twitter application, although the media criticized the decision.[170]

In this way, monopolists of software products may evade competitors' efforts to gain access to essential content or products without interfering with their own users' experience or producing new generations of products that require users to make a switch from older models. This is vital because it frustrates the courts' market-preference test designed to identify bona fide innovation.[171] Users of a competitor's product that pulls heavily from information accessed through the API of a dominant firm's product may not have the opportunity to assess whether they prefer the dominant firm's new product. Instead, consumers are forced to use the dominant firm's product instead of the one they had been using, with no discernable change in the content or presentation that consumers consider when choosing which product to use.

Thus, software products differ markedly from the cases involving physical products that gave rise to current predatory innovation law. Unlike markets in physical products, software markets are more likely to be defined by network effects, allow for the elimination of older products, and permit anticompetitive innovation without any user-facing changes. These factors suggest that the existing law of predato-

[169] *See* Garrett, *supra* note 120 and accompanying text.

[170] *See supra* notes 16–19 and accompanying text; *see also* Charles Arthur, *Twitter Token Limit Criticised as Falcon Pro Hikes App Price To Deter Buyers*, GUARDIAN (Feb. 27, 2013, 7:49 AM), http://www.theguardian.com/technology/appsblog/2013/feb/27/twitter-token-limit-falcon-pro [http://perma.cc/4R3Q-E973]; Casey Chan, *Twitter Wants To Stop You from Using Twitter Apps Not Made by Twitter*, GIZMODO (Aug. 16, 2012, 6:50 PM), http://gizmodo.com/5935517/twitter-wants-to-stop-people-from-using-apps-not-made-by-twitter [http://perma.cc/3N3P-84J8] ("User caps for third-party Twitter apps will be based on the current number of users that the third-party apps already have. For x amount of users, a third-party Twitter app can now only support twice as much, or 2x, amount of users for the rest of the app's existence.").

[171] *See supra* Part III.B.3.

ry innovation — which largely depends on gauging market acceptance — may not apply easily to software products.

V. Tests for Tying, Unilateral Refusals To Deal, and Essential Facilities Are Inapplicable or Insufficient

Before elaborating a new test for predatory innovation products, it is necessary to address why software redesign may not be reached by other established tests for anticompetitive conduct: tying, unilateral refusals to deal, and essential facilities. Because our focus is on breaks in interoperability, it would seem that a monopolist would be liable under a theory of unilateral refusals to deal. Similarly, because several of the breaks of interoperability include efforts by a monopolist to introduce a new product, the product changes could plausibly raise concerns of tying. As described below, these theories may reach some breaks in interoperability, but not all. Ultimately, monopolization in software markets extends beyond either tying or unilateral refusals to deal and warrants a new test for predatory innovation.

A. Tying

A tying arrangement is when a firm with market power[172] in one market attempts to force consumers to buy a related, non-monopolized product by "tying" it to the dominant product.[173] Tying arrangements limit free consumer choice by requiring them to purchase the tied product to gain access to the dominant tying product, such as when the manufacturer of a printer requires printer cartridges to be purchased from the same manufacturer. To prevail on a tying claim, a plaintiff must show (1) that the defendant has market power over the tying product, (2) that the consumer is forced to purchase a second product in order to procure the first, and finally (3) that the tie has foreclosed a "not insubstantial amount of commerce."[174] Courts have variously identified the test as having three or four elements,[175] the fourth being (4) that implicitly or expressly, the forced tie must be by contract.[176] Consequently, a "technological tie" in which the domi-

[172] As noted above, although tying is technically a claim of collusive conduct under section 1 of the Sherman Act, it requires a showing of market power, Jefferson Par. Hosp. Dist. No. 2 v. Hyde, 466 U.S. 2, 33 (1984), and thus we treat it as closely related to monopolization under section 2 of the Act.

[173] Foremost Pro Color, Inc. v. Eastman Kodak Co., 703 F.2d 534, 540 (9th Cir. 1983); *see also* Digidyne Corp. v. Data Gen. Corp., 734 F.2d 1336, 1344 (9th Cir. 1984) (affirming jury verdict that operating system monopolist's refusal to sell its operating system to manufacturers not using its CPUs constituted tying).

[174] *Jefferson Par. Hosp.*, 466 U.S. at 8.

[175] Thomas H. Au, *Anticompetitive Tying and Bundling Arrangements in the Smartphone Industry*, 16 STAN. TECH. L. REV. 188, 204 (2012).

[176] *Id.*

nant product in one market is designed to work only with a specific, peripheral product does not give rise to an unlawful tying arrangement.[177]

Tying may not reach all cases where breaks in interoperability pose threats to competition for at least three reasons. First, predatory innovation in software markets may not even involve a second, tied product. Twitter's limitation of user tokens for third-party applications is an example of this. Twitter effectively barred third-party applications from competing for additional shares of the user base.[178] The limitation on user tokens, however, did not tie two products together, even though it forced Twitter's user interface on consumers. Without tying two products together, breaks in interoperability that exclude competitors cannot give rise to a tying claim.

Second, even if the break in interoperability serves to tighten the connection between two of the monopolist's products, the connection may not amount to a forced sale. Facebook's acquisition of and integration with Instagram highlights this. Prior to Facebook's acquisition of the photo sharing company, photos taken on the Instagram application could appear "in line" on Twitter. Following the acquisition, however, Facebook eliminated this feature for Twitter, while retaining it for the primary Facebook network.[179] Although Facebook made it far more appealing for users to post to Facebook rather than Twitter, the change seems unlikely to constitute a forced sale; Instagram users may still post links to their photos on Twitter, and — vitally — are not required to post their content to Facebook at all.

Even Apple's manipulation of DRM and iTunes may not constitute tying per se, as users could use the iPod for music burned from compact discs or enjoy music purchased from the iTunes Store on their computers without an iPod. In both cases, the close tie between products plausibly discouraged, but did not foreclose, user choice among products; because there is no contractual obligation for the iTunes or Facebook users to use an iPod or Instagram, the scenarios here are unlikely to fulfill the contractual forced sale element of a tying claim.

As described above, however, this type of conduct still poses a threat to competition, despite not meeting the requirements for a tying claim. Twitter, with complete control over its content network, limited third-party access to reduce competition on mobile platforms. Likewise, Facebook, recognizing the importance of photo sharing for social networking, precluded its chief competitor from accessing one of the most popular photo sharing applications available. The limited scope of tying claims, coupled with the weakness of current predatory

[177] *Foremost Pro Color, Inc.*, 703 F.2d at 542.

[178] *See* Chan, *supra* note 170.

[179] *See supra* notes 21–25 and accompanying text.

innovation doctrine, makes it unlikely that similar efforts at exclusion will be subject to any antitrust scrutiny.

Finally, the difficulty of reverse engineering software products is another reason why tying is insufficient to reach many breaks in interoperability. In traditional technological tying cases, competitors may use reverse engineering to work around the predatory design and, as noted above, courts have often relied on this in finding defendants not liable. This is especially true where the tied product is in a peripheral market. Courts considering peripheral product cases often characterize the dominant firm not as monopolistic, but simply as having a head start — the benefit of its investment in research and development. Software products, however, can be harder to reverse engineer, and may even explicitly block access by particular (or all) competitors.[180] Consequently, the rationale traditionally supporting deference to product changes is even less applicable in a software market, where breaks in interoperability are not a head start that may be engineered around, but an absolute bar to competition.

B. Unilateral Refusals To Deal

By their nature, breaks in interoperability exclude competitors from access to certain resources of the dominant firm, and such conduct seems to naturally give rise to claims of a unilateral refusal to deal. A unilateral refusal to deal is when a single monopolist refuses to cooperate with another company, usually a competitor.[181] The bar for such a claim currently appears to be high, although the case law has varied over time in a myriad of factual situations. Courts generally presume that a monopolist's refusal to deal is justified,[182] but there are exceptions. One, set forth in *Lorain Journal*, occurs when the refusal to deal serves no purpose but to exclude competition.[183] One specific and oft-cited permutation of the *Lorain* exception is when a monopolist was previously engaged in a profitable arrangement with a com-

[180] *See supra* notes 127–33 and accompanying text.

[181] United States v. Colgate & Co., 250 U.S. 300, 307 (1919).

[182] *See* Image Tech. Servs. v. Eastman Kodak Co., 125 F.3d 1195, 1216 (9th Cir. 1997); CSU, L.L.C. v. Xerox Corp. (*In re* Indep. Serv. Orgs. Antitrust Litig.), 203 F.3d 1322, 1328–29 (Fed. Cir. 2000); Data Gen. Corp. v. Grumman Sys. Support Corp., 36 F.3d 1147, 1155 (1st Cir. 1994). This is unlike concerted refusals to deal by competitors acting in a cartel, which are illegal per se. *See, e.g.*, Jones Knitting Corp. v. Morgan, 361 F.2d 451, 459 (3d Cir. 1966) (holding as illegal per se an agreement among accused infringers not to settle with a plaintiff); *cf.* Gould v. Control Laser Corp., 462 F. Supp. 685, 691–93 (M.D. Fla. 1978) (treating such an agreement under the rule of reason).

[183] Lorain Journal Co. v. United States, 342 U.S. 143, 153–55 (1951) (holding that newspaper with ninety-nine percent market share violated the Sherman Act when it refused to sell advertisements to businesses that also advertised on a newly opened radio station).

petitor and then changed course, refusing to deal with no apparent purpose but to drive the competitor from the market.[184]

While refusal to license intellectual property rights has not been deemed an antitrust violation in the United States, courts have ordered antitrust violators to license those rights and disclose nonpublic information, such as interface specifications, to competitors.[185] For example, the consent decree settling the antitrust case against Microsoft in the 1990s required Microsoft to disclose interface information and license intellectual property to firms developing interoperable technologies.[186] The court imposed these terms as necessary to restore competition, despite the tenuous at best connection to illegal conduct — antitrust authorities had not charged Microsoft with misusing patents on interfaces or refusing to license intellectual property rights in those interfaces to competitors.[187] The European Commission also compelled Microsoft to disclose specifications that would allow competitors to create interoperable products with Windows technologies.[188]

Because of the complexities of breaks in interoperability in software products, the logic behind exclusive dealing cases is likely to be inapplicable. In *Aspen Skiing*, for example, three independent ski companies operated the four ski areas in Aspen, Colorado.[189] One of the companies, Ski Co., acquired three of the four resorts.[190] With only one remaining competitor, Ski Co. discontinued the long-standing practice of all four ski areas to jointly issue an "all-Aspen" pass.[191] In evaluating whether Ski Co.'s conduct violated section 2 of the Sherman Act, the Supreme Court noted that "[i]f a firm has been 'attempting to exclude rivals on some basis other than efficiency,' it is fair to characterize its behavior as predatory."[192] Ski Co.'s conduct amounted to more than a refusal to deal with a rival, but rather eliminated a long-standing, profitable practice.[193] Ski Co.'s change in the market eliminated a superior product preferred by consumers and hin-

[184] *See* Aspen Skiing Co. v. Aspen Highlands Skiing Corp., 472 U.S. 585, 602–03 (1985).

[185] *See*, *e.g.*, William E. Kovacic, *Designing Antitrust Remedies for Dominant Firm Misconduct*, 31 CONN. L. REV. 1285, 1304–05 (1999) (giving examples of licenses and disclosures induced or compelled by antitrust authorities).

[186] *See* William H. Page & Seldon J. Childers, *Software Development as an Antitrust Remedy: Lessons from the Enforcement of the Microsoft Communications Protocol Licensing Requirement*, 14 MICH. TELECOMM. TECH. L. REV. 77, 83, 89–90 (2007) (describing these "forward-looking" provisions).

[187] *See id.*

[188] *See* Samuelson, *supra* note 37, at 1987–2001 (discussing competition and antitrust law used to facilitate interoperability generally).

[189] *Aspen Skiing*, 472 U.S. at 589.

[190] *Id.* at 589–90.

[191] *Id.* at 592.

[192] *Id.* at 605 (quoting ROBERT BORK, THE ANTITRUST PARADOX 138 (1978)).

[193] *Id.* at 603.

dered its competitors' ability to compete, and Ski Co. did not offer an efficiency-enhancing explanation for its conduct.[194] Consequently, the Court held that a jury could reasonably infer that the elimination of the all-Aspen pass was exclusionary.[195]

Aspen Skiing, along with its predecessor *Lorain Journal*,[196] marks the courts' willingness to curtail a monopolist's right to refuse to deal with a rival where that exclusion is "attempting to exclude rivals on some basis other than efficiency."[197] The complexities of breaks in interoperability suggest that a monopolist could always supply such a justification, especially when the dominant firm implements the break simultaneously with other changes.[198] Consequently, defendants may readily point to a business justification to explain their actions.[199]

Further — and perhaps more importantly — if the dominant firm has copyright or patent protection for its product, courts generally presume that excluding rivals from the protected work is a valid business justification.[200] The presumption of validity may be rebutted by showing that the patent or copyright was acquired in an "unlawful manner,"[201] that the protection of the patent or copyright is a pretext,[202] or that the protection was obtained by "illegal tying, fraud in the Patent and Trademark Office, or sham litigation."[203]

In *Data General Corp. v. Grumman System Support Corp.*, the First Circuit considered whether Data General's refusal to license its diagnostic software to "third-party maintainers" constituted exclusionary conduct. Data General had previously encouraged liberal use of its diagnostic tools by third parties before it began limiting access in order to maximize profits in the service aftermarket.[204] Because Data General denied the third parties access to its software under copyright law, the exclusion was presumably a valid business justification.[205] Other courts have gone so far as to reject any rebuttal other

[194] *Id.* at 604–10.

[195] *Id.* at 610–11.

[196] Lorain Journal Co. v. United States, 342 U.S. 143, 154 (1951).

[197] *Aspen Skiing*, 472 U.S. at 605 (quoting BORK, *supra* note 192, at 138).

[198] Somers v. Apple, Inc. (*In re* Apple iPod iTunes Anti-Trust Litig.), Nos. C 05-00037 JW, C 07-06507 JW, 2010 WL 2629907, at *7–8 (N.D. Cal. June 29, 2010).

[199] The procompetitive justification offered by a defendant is open to rebuttal by the plaintiff. *Id.* The weighing of the two is a question of fact. *Id.*

[200] Data Gen. Corp. v. Grumman Sys. Support Corp., 36 F.3d 1147, 1187 (1st Cir. 1994); *see also* CSU, L.L.C. v. Xerox Corp. (*In re* Indep. Serv. Orgs. Antitrust Litig.), 203 F.3d 1322, 1329 (Fed. Cir. 2000); Image Tech. Servs. v. Eastman Kodak Co., 125 F.3d 1195, 1218 (9th Cir. 1997).

[201] *Data Gen. Corp.*, 36 F.3d at 1188.

[202] *Image Tech. Servs.*, 125 F.3d at 1219. Other courts have explicitly rejected the role of intent in rebutting the presumption of valid exclusion from a patent or copyright. *See, e.g.*, *Xerox*, 203 F.3d at 1329.

[203] *Xerox*, 203 F.3d at 1327.

[204] *Data Gen. Corp.*, 36 F.3d at 1154.

[205] *Id.* at 1187.

than acquisition of the patent or copyright by fraud or illegal conduct before the relevant authority.[206]

Even *Aspen Skiing* did not apply to the facts here. "Apparently sensing the uphill nature of its allegation of an exclusionary refusal to license," the third-party maintainer argued that Data General's conduct was covered by *Aspen Skiing*.[207] Even if an *Aspen Skiing* claim *could* have overcome the presumption of a valid copyright exclusion, the court rejected its application to the case.[208] Noting that *Aspen Skiing* rested on a comparison of the monopolist's behavior in competitive and monopolized markets, the court rejected the analogy, as Data General had always been a monopolist.[209]

The protections afforded patent and copyright owners make it difficult to bring a successful unilateral refusal to deal claim. Software products are often protected by both patent and copyright. Under *Data General* and *Xerox*, these protections are likely to be recognized as valid business justifications for exclusion.[210] Further, breaks in interoperability are not always accompanied by a change in the nature of the market to allow a comparison of the dominant firm's conduct in both competitive and monopolized markets.[211] Given the courts' reluctance to find liability for refusals to license software in *Data General* and *Xerox*, such an argument seems unlikely to succeed against a monopolist breaking interoperability for software products.

C. Essential Facilities

Another exception to permissibility of a unilateral refusal to deal exists, albeit extremely rarely. Where a monopolist controls an "es-

[206] The Federal Circuit has explicitly limited rebuttal of the presumption to traditional exceptions to the monopolies afforded by intellectual property protections: patent misuse, sham litigation, and *Walker Process* claims. *Xerox*, 203 F.3d at 1326–28; *see also id.* at 1329 ("[I]n the absence of any evidence that the copyrights were obtained by unlawful means or were used to gain monopoly power . . . . Xerox's refusal to sell or license its copyrighted works was squarely within the rights granted by Congress to the copyright holder . . . .").

[207] *Data Gen. Corp.*, 36 F.3d at 1188.

[208] *Id.*

[209] *Id.*

[210] In the Ninth Circuit, however, plaintiffs may rebut the presumption of a valid business justification by showing that the exclusion from the patent was a mere pretext. Image Tech. Servs. v. Eastman Kodak Co., 125 F.3d 1195, 1219 (9th Cir. 1997). The Ninth Circuit's refusal to reverse the judgment for the plaintiff was supported by testimony by key Kodak decision makers that protection of patent and copyrights had never occurred to them. *Id.* Further, only sixty-five of the thousands of parts at issue in *Image Tech. Servs.* were protected by patents; consequently, a jury was more likely than not to find that patent protection was a pretext for its exclusionary activity. *Id.* at 1219–20. This situation differs from that of software, in which the entire product is likely to be protected by copyright or patent rights.

[211] *Data Gen. Corp.*, 36 F.3d at 1188.

sential facility"[212] — one that is necessary for a competitor to enter and compete in the market — and the competitor cannot replicate the facility, a court may find that the monopolist has a duty to deal.[213] The essential facilities doctrine works best as applied to historical and modern infrastructure not easily replicated by competitors, such as railroad bridges, seaports, or telephone networks.[214] The doctrine has more recently been invoked in Europe to analyze other types of facilities, including information products, software, and even interface specifications.[215] However, it may not (easily) extend to non-physical facilities in the United States,[216] although commentators have encouraged its application to software products, such as unregulated software platforms.[217]

To bring an essential facilities challenge to a unilateral refusal to deal, a plaintiff must show four elements: (1) control of the essential facility by a monopolist; (2) a competitor's inability — practically or reasonably — to replicate the essential facility; (3) denial of the use of the facility to a competitor; and (4) feasibility of providing the facility.[218] Courts have generally required that the facility in question be in an upstream market and that the monopolist deny access to the facility solely to eliminate competition in the downstream market.[219]

When pursuing a claim of a break in interoperability against a dominant firm, it is especially difficult to establish the inability to replicate the facility. The bar for "inability practically or reasonably to replicate the essential facility" is high, often thought to be limited to "facilities that are a natural monopoly, facilities whose duplication is forbidden by law, and perhaps those that are publicly subsidized and

[212] Although some courts and commentators have included *Aspen Skiing* in the genealogy of the essential facilities doctrine, the Supreme Court has indicated that it considers the two to be distinct theories. *See* Verizon Commc'ns Inc. v. Law Offices of Curtis V. Trinko, LLP, 540 U.S. 398, 410–11 (2004) ("We have never recognized such a doctrine, and we find no need either to recognize it or to repudiate it here." (citations omitted)). Lower courts have similarly assumed that the theory of liability presented by *Aspen Skiing* is distinct from that developed in the essential facilities cases. *See Data Gen. Corp.*, 36 F.3d at 1183–84 ("It is not entirely clear whether the Court in *Aspen Skiing* merely intended to create a category of refusal-to-deal cases different from the essential facilities category or whether the Court was inviting the application of more general principles of antitrust analysis to unilateral refusals to deal.").

[213] *See Verizon Commc'ns Inc.*, 540 U.S. at 411.

[214] *See* Intergraph Corp. v. Intel Corp., 195 F.3d 1346, 1356 (Fed. Cir. 1999).

[215] James Turney, *Defining the Limits of the EU Essential Facilities Doctrine on Intellectual Property Rights: The Primacy of Securing Optimal Innovation*, 3 NW. J. TECH. & INTELL. PROP. 179, 192 (2005).

[216] *See Intergraph Corp.*, 195 F.3d at 1357–58 (holding that the essential facilities doctrine did not apply to providing technical data to a non-competing entity).

[217] Sandeep Vaheesan, *Reviving an Epithet: A New Way Forward for the Essential Facilities Doctrine*, 2010 UTAH L. REV. 911, 951–59 (2010) (urging application of essential facilities to unregulated software platforms).

[218] MCI Commc'ns Corp. v. Am. Tel. & Tel. Co., 708 F.2d 1081, 1132–33 (7th Cir. 1983).

[219] *Intergraph Corp.*, 195 F.3d at 1357.

thus could not practicably be built privately."[220] Thus, successful essential facilities claims have generally involved extremely expensive facilities or those underlying natural monopolies such as entire power grids, mountains, or sports complexes.[221] In other successful challenges, "the facility in question was more than dominant; it was effectively the only one in town."[222]

In the software context, what would constitute an "essential facility" is unclear. Depending on how an essential facilities claim is framed, a court may not view a break in interoperability as an antitrust violation because the facility is replicable or not necessary to compete in the market.

When Apple introduced its FairPlay DRM technology, competitors could argue that the lost facility was ability to play music purchased from their music stores; however, that function was clearly replicated, at least temporarily, when RealNetwork responded with its Harmony technology, which allowed RealNetwork music to be played on Apple digital music players.[223] Similarly, when Instagram ceased to allow photos to be posted to Twitter in 2012,[224] Twitter might have argued that the essential facility it lost was access to Instagram content and photo sharing; however, Twitter replicated the feature with its own robust photo sharing features,[225] and users could upload photos to both platforms. Although Twitter did not (and could not) reproduce the whole network, its own photo sharing features may indeed be a sufficient replication of the lost facility.[226] These examples suggest that in the software world, where readily understood physical limitations on replication are not present, facilities may *not* be "more than dominant." A court may rule that comparable "facilities" can be reproduced.

A plaintiff challenging a break in interoperability may argue that, although it can create a similar software product, network effects pre-

[220] Twin Labs., Inc. v. Weider Health & Fitness, 900 F.2d 566, 569 (2d Cir. 1990) (quoting PHILLIP E. AREEDA & HERBERT HOVENKAMP, ANTITRUST LAW ¶ 736.1, at 655 (Supp. 1988)).

[221] *Id.*

[222] *Id.*

[223] *See* Singleton & Lowensohn, *supra* note 10.

[224] *See* Rao, *supra* note 22.

[225] *See* Chris Taylor, *Twitter Now Fills Your Feed with Photos*, MASHABLE (Oct. 29, 2013), http://mashable.com/2013/10/29/twitter-photos-videos/ [http://perma.cc/KTC9-S6PW].

[226] Although it may seem odd for the defendants to argue that interoperability of photo sharing communities constitute lost "facilities," classic cases at the foundation of the doctrine similarly stray far from the quintessential examples of power grids or sports complexes. *See, e.g.*, Aspen Skiing Co. v. Aspen Highlands Skiing Corp., 472 U.S. 585, 608 (1985) (holding the defendant was required to continue a profitable agreement with the plaintiff); Otter Tail Power Co. v. United States, 410 U.S. 366, 377 (1973) (affirming district court's order requiring a defendant power company to sell power wholesale to municipal systems); Lorain Journal Co. v. United States, 342 U.S. 143, 149–50 (1951) (holding the defendant was obligated to sell advertising space).

vent the new "facility" from having similar impact.[227] Essential facilities claims are usually leveraged against natural monopolies,[228] which have supply side economies of scale.[229] These traditional essential facilities are usually marked by some physical or geological feature that makes entrance by new competitors cost prohibitive.[230] Network effects, however, operate on the demand side, reflecting increased value to consumers with each additional user.[231] New entrants cannot provide the same value as current firms, even with an identical product, because the user base is part of the value. Thus Twitter could instead argue that the lost facility was not just the Instagram photos themselves, but the entire Instagram community. A facility defined broadly to encompass this community of users could not simply be recreated by engineers; a community is not practical or reasonable to replicate, due in part to network effects acting as a barrier to entry. Nonetheless, because software is so readily distinguishable from the natural monopolies where the doctrine emerged, it is not clear that the essential facilities doctrine would apply to software products.

VI. Structured Rule of Reason Test for Software Products

Under the current focus of predatory innovation on the "no business justification" standard, courts may too readily defer to defendants' alleged business justifications to address anticompetitive harm from software product design changes. The examples of Twitter and Apple demonstrate the ease with which a dominant firm may exclude competitors it once benefited from, without fear of antitrust liability. Software products have several characteristics that distinguish them from physical products and make them uniquely able to use product redesign to break interoperability. Firms can implement and distribute changes in software quickly, such as in response to competitor behavior. Further, the corresponding software product markets defy assumptions underlying predatory innovation analysis for physical product markets, rendering the doctrine of predatory innovation largely ineffective.

Users may have no realistic opportunity to choose whether to adopt the new product. Consequently, the changes in a software product are particularly apt to exclude competition by avoiding the characteristic that assuaged courts in earlier cases: reverse engineering, the maintenance of older products, and the market's election of the

[227] *See* Lemley & McGowan, *supra* note 7, at 496.

[228] *See* Speta, *supra* note 27, at 1280.

[229] *Id.* at 1279–80.

[230] *Id.* at 1280.

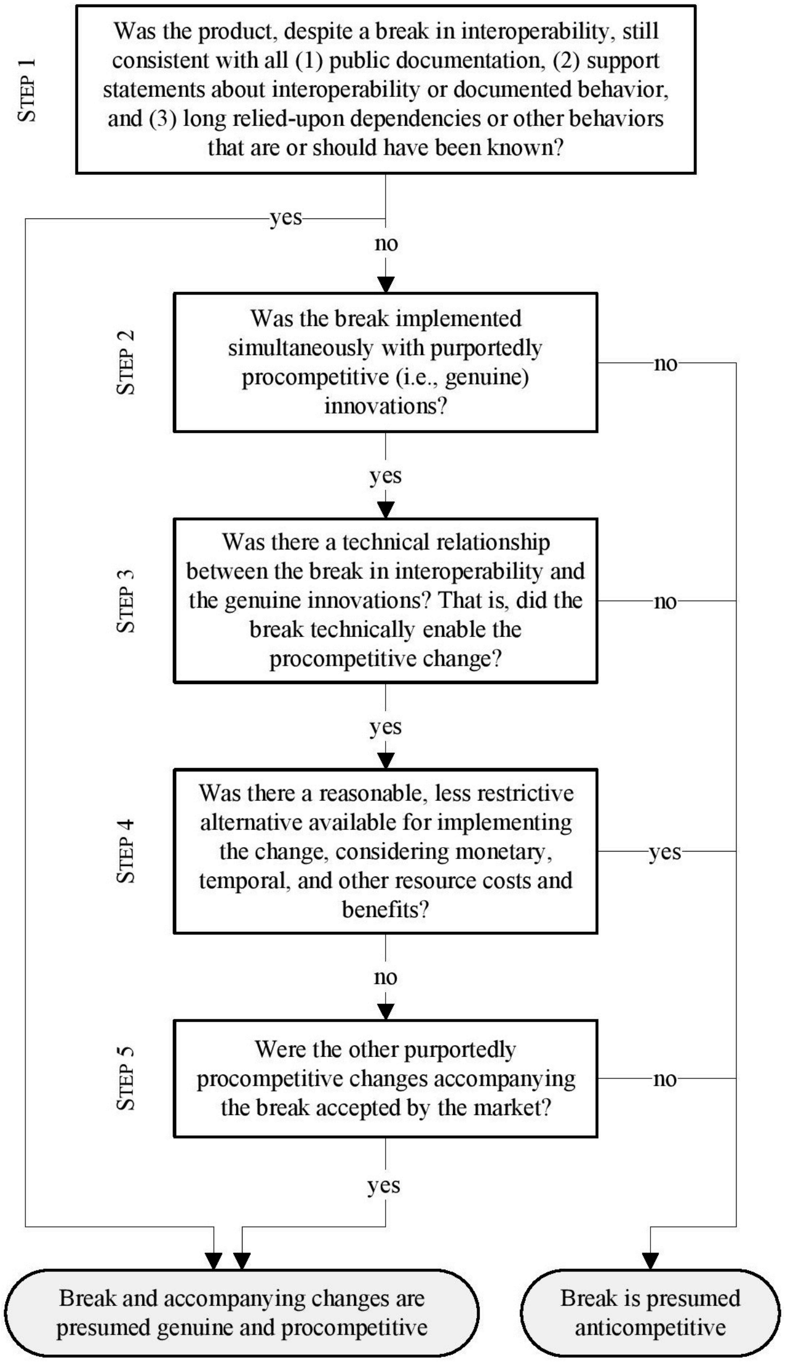
[231] *Id.*

new product (and the inference that it is indeed an "innovation"). Without these characteristics, producers of software products are able to exclude competitors to a degree not paralleled in markets for physical products.

Antitrust law should carefully examine when software product redesign that results in non-interoperability constitutes anticompetitive conduct. We propose a framework for a test that addresses these differences: a structured rule of reason test for software products. A rule of reason analysis would emphasize weighing procompetitive and anticompetitive effects, rather than allowing a firm to escape liability for any degree of anticompetitive effect so long as there is some procompetitive justification, as evidenced by market acceptance or shown by a business justification. The proposed test would (1) help prevent dominant firms from taking advantage of network effects to the detriment of partners who helped establish broad usage, in reliance on support statements; (2) consider whether the innovation necessarily led to the elimination of interoperability and the viability of other products; (3) address whether the change influenced the consumer experience and was meaningfully accepted by consumers; and (4) utilize the court's procedural rather than technical competence.

In particular, this framework considers the competence of courts to scrutinize software and redesigns, while avoiding over-deference to market acceptance. The framework's guidance helps to weigh procompetitive and anticompetitive effects in this technical industry, where courts and agencies may be especially wary of the likelihood of errors in analysis and their potential consequences. The framework also bears in mind strong policy reasons to minimize interference with software product markets, where overactive judicial interference may inhibit innovation and competition. In doing so, it attempts to provide structure to make outcomes more predictable, to aid both courts in applying, and companies in complying with, the proposed standard.

The test is sequential in nature, guiding courts through "easier" or more critical questions first, in order to dispose of predatory innovation challenges earlier in the analysis. This improves the predictability of outcomes by reserving the balancing factors and more technical analysis for later steps. Thus, we have refined the test as follows:


STEP 1
Was the product, despite a break in interoperability, still consistent with all (1) public documentation, (2) support statements about interoperability or documented behavior, and (3) long relied-upon dependencies or other behaviors that are or should have been known?
yes
no
STEP 2
Was the break implemented simultaneously with purportedly procompetitive (i.e., genuine) innovations?
no
yes
STEP 3
Was there a technical relationship between the break in interoperability and the genuine innovations? That is, did the break technically enable the procompetitive change?
no
yes
STEP 4
Was there a reasonable, less restrictive alternative available for implementing the change, considering monetary, temporal, and other resource costs and benefits?
yes
no
STEP 5
Were the other purportedly procompetitive changes accompanying the break accepted by the market?
no
yes
Break and accompanying changes are presumed genuine and procompetitive
Break is presumed anticompetitive

1. Was the product, despite a break in interoperability, still consistent with all (1) public documentation, (2) support statements about interoperability or documented behavior, and (3) long relied-upon dependencies or other behaviors that are or should have been known?
   - If not, move on to step 2.
   - If so, the break is presumed genuine and procompetitive.
2. Was the break in interoperability implemented simultaneously with purportedly procompetitive (i.e., genuine) innovations?[232]
   - If not, the break is presumed anticompetitive.
   - If so, move on to step 3.
3. Was there a technical relationship between the break in interoperability and the genuine innovations? That is, did the break in interoperability technically enable the procompetitive change?[233]
   - If not, the break is presumed anticompetitive.
   - If so, move to step 4.
4. Was there a reasonable, less restrictive alternative available for implementing the change? That is, was the break in interoperability a necessary means of achieving the other changes, or was the break reasonable considering monetary, temporal, and other resource costs and benefits of alternatives?
   - If so, the break is presumed anticompetitive.
   - If not, move to step 5.
5. Were the other purportedly procompetitive changes accompanying the break in interoperability accepted by the market?
   - If not, the break is presumed anticompetitive.
   - If so, the break and the accompanying changes are presumed genuine and procompetitive.

[232] The timing of conduct has weighed in the analysis of monopolization in other cases. *See, e.g.*, Berkey Photo, Inc. v. Eastman Kodak Co., 603 F.2d 263, 286 (2d Cir. 1979). Courts have been willing to find monopolists liable for timing changes in order to exclude competitors under regulatory regimes, such as the Hatch-Waxman Act. *See* Abbott Labs. v. Teva Pharm. USA, Inc., 432 F. Supp. 2d 408, 422 (D. Del. 2006). Here, the timing serves not to find liability, but as a logical necessity to the analysis: where the alleged procompetitive change is implemented at a time removed from the break in interoperability, the two are unlikely to be related.

[233] This may be the type of explanation provided by Google when it moved Hangouts away from XMPP. *See supra* notes 131–32132 and accompanying text.

The test embodies two basic principles. First, it avoids any mandate of interoperability[234] by presuming anticompetitive conduct only if the non-interoperability differs from what the firm has promised to developers of complementary or competing products through documentation, support statements, or long held practice. In some sense, the sequential test here serves as a structured rule of reason. It guides the court through relevant considerations in determining whether the defendant's conduct was exclusionary, but allows the defendant to provide an exculpatory justification outside its rigors.[235] This recognizes the complexities of software markets in particular, providing structure to the court's reasoning without artificially ruling out an opportunity for the defendant to carry her burden. Second, the test escalates in degree of technical sophistication necessary to answer each successive question. It is easier for a court to determine if a break in compatibility occurred at the same time as other changes than it is to determine if the break was "necessary" for the other changes, which would presumably be a battle of experts.

The test here is designed to isolate the third factor that distinguishes software markets from physical ones: interoperability and other aspects of the product may be altered with minimal or no change to the experience for current users and advertisers, but with crippling effects for competitors. As described above, this characteristic means that courts cannot rely on the market to approve a change as a genuine innovation, as users and advertisers may be forced to migrate without seeing any corresponding change in the underlying product. This test helps courts determine whether the innovation was genuine in the software context, where the reason for the shift in the market (from third-party applications to Twitter, for example) may not be so obviously motivated by product features (as opposed to *Berkey Photo*, where users were presumably buying the new camera and film for the features of those new products).

The first step of the test considers that developers may rely on promises of support and established interoperability[236] and protects firms whose redesign is consistent with past conduct and support statements. A plaintiff may argue, as in *Aspen Skiing*, that the elimination of a profitable business arrangement (interoperability) only after the defendant becomes dominant is likely to serve only exclusionary purposes. If the defendant has clearly stated how it will or will not support interoperability, such as with Facebook's public documenta-

[234] *See supra* notes 4–5 and accompanying text (discussing mandated interoperability and costs thereof).

[235] Such balancing is common in antitrust analysis and well within the competency of the courts. *See* C. Scott Hemphill, Less Restrictive Alternatives in Antitrust and Constitutional Law (Feb. 26, 2015) (unpublished manuscript), http://scholarship.law.berkeley.edu/cgi/viewcontent.cgi?article=1031&context=law_econ [http://perma.cc/W74S-BQQ4].

[236] *See supra* Parts IV.A–B.

tion and versioning for APIs, such a reliance argument is weak. Adherence to published or long-established standards suggests the defendant did not suddenly change its business practices, but instead was adhering to them. Consequently, adherence to documentation or industry standards establishes a presumption that the change was justified, leaving the plaintiff to establish exclusionary conduct by other means.

The second step focuses on whether the allegedly anticompetitive effect was contemporaneous with any procompetitive justification offered by the defendant. If the procompetitive improvement is far removed in time from the break in interoperability, it seems unlikely that the two are actually related. Thus, if Twitter limited user tokens for mobile applications[237] long before implementing any interface changes to improve the mobile experience, the court may presume that the justification is a sham and unrelated to the break. The burden would shift to the defendant to show that, despite the remoteness in time, the break in interoperability was tied to the procompetitive innovation.

Establishing the tie comes in the third step, where the court considers whether the break of interoperability was technically related to the procompetitive change or improvement. The focus here is not on whether the break was *necessary* for the procompetitive change, but on whether there is some technical connection linking the non-interoperability and procompetitive innovation. Again, the idea of the sequential test is rooting out undeniably exclusionary breaks with no procompetitive justification before reaching more difficult technical questions. If there is no apparent technical relation, the defendant would bear the burden of showing some other compelling procompetitive justification.

The final two steps embody the more technical questions. In step four, the court must determine whether the defendants had a reasonable and less restrictive alternative to implementing the procompetitive change. This step comprises two questions: First, was there a less restrictive alternative that would enable the procompetitive change without breaking interoperability? Second, would that alternative be reasonable to implement considering, among other things, technical complexity, impact on product release schedule, and maintainability?[238] As pointed out by C. Scott Hemphill, the less restrictive alternative analysis often serves to "smoke out" anticompetitive effect.[239] The assessment forces the defendant to explain why the break in interoperability — which may harm competition — is necessary for the

[237] *See supra* Part II.A.

[238] At this point, step three of the test should have already determined that the break in interoperability was related to — and in some sense, enabled — the procompetitive change.

[239] *See* Hemphill, *supra* note 235, at 45–46.

purported procompetitive change implemented by the defendant.[240] If the defendant cannot show that the break in interoperability is necessary to some procompetitive benefit, the court should presume the break is anticompetitive. If the defendant can show the break in interoperability is necessary, the court proceeds to step five. Admittedly, the step is technical and thus may require experts to determine whether a less restrictive alternative is available.[241]

In the final step, the defendant must demonstrate that the non-interoperability changes were indeed legitimate innovations. The test turns to whether the market accepted the change accompanying the break in interoperability. This may seem incongruous given our rejection of the market acceptance test espoused by the Second Circuit.[242] The focus here, however, is no longer on the break in interoperability — which may indeed occur beneath the market's notice — and is instead on the accompanying procompetitive change offered by the defendants. The idea is that even if the break in interoperability is reasonably necessary for the change, that change is procompetitive only if accepted by the market. The goal is to preclude defendants from breaking interoperability and supplying a sham "improvement" visible to the market with the primary purpose of excluding competition. Thus, the final step of the test adopts the Second Circuit's test in *Berkey Photo*, but only for those purportedly procompetitive, market-visible changes accompanying the break in interoperability.

VII. CONCLUSION

This Note demonstrates how current antitrust doctrines fail to curb anticompetitive redesign of software products that break interoperability and suggests a new framework to evaluate those changes. While courts and agencies may be rightly concerned about in-depth technical analysis of each line of code and the reasons behind each change, the proposed framework provides a basis for analysis within their competence, creating a sequential structure beginning with bright-line questions and finishing with a more technical analysis. This framework allows courts to evaluate firm conduct and product changes in light of the unique characteristics of software markets.

[240] *See id.* at 46.

[241] Although step four (and in turn, step five) presents a more technical question, technical analysis is still within the competence of the courts. Courts regularly tackle questions regarding least restrictive alternatives, Hemphill, *supra* note 235, at 7–8, complicated issues of market definition, *see* Jefferson Par. Hosp. Dist. No. 2 v. Hyde, 466 U.S. 2, 34 (1984) (O'Connor, J., concurring) (discussing the "elaborate inquiry into the economic effects" required for market definition and market power analyses), and the technical aspects of patent claims, *cf.* Markman v. Westview Instruments, 517 U.S. 370, 388 (1996) (discussing the competence of the court to interpret patent claims).

[242] *See supra* Part III.B.