Notes

Competitive Regulation of Mobile Software Systems: Promoting Innovation Through Reform of Antitrust and Patent Laws

JOSH BASKIN*

This Note analyzes the current antitrust regulatory framework for high tech, iteratively evolving computer and software systems. This issue has significant implications for the current economy as many modern technology companies base their entire business model on such systems. This Note examines the problems concerning software patents through an analysis of two well-known mobile phone operating systems: Apple's iOS and Google's Android.

This Note also examines the current regulatory framework that prevents large companies from taking anticompetitive actions to expand their power in fast-moving high tech markets at the expense of smaller competitors—specifically tying, predatory innovation, refusal to deal or license, sham litigation, and overbroad software patents. This Note also proposes several changes to both antitrust and patent laws that will make it more difficult for established market players to prevent new competitors from entering high tech markets, thereby promoting greater openness and innovation. These changes include modernizing sham litigation, reducing the number of patent infringement actions by allowing reverse engineering of software patent and an independent invention defense, and increased scrutiny of the business improvements antitrust defense.

Each of these proposed changes targets the promotion of innovation by enabling the entry of new players into established markets without the threat of expensive litigation constantly undermining the compatibility and efficiency of the products that they attempt to bring to market.

* J.D., University of California, Hastings College of the Law, 2013. B.A. University of California, Berkeley, Economics, 2010. I extend my thanks to the editors of the *Hastings Law Journal* for their suggestions, support, and editorial assistance. I would also like to thank Professor Dorit Reiss for her suggestions and feedback throughout the editing process.

Table of Contents

Introduction

Innovation defines the high tech industry. To survive and thrive, high tech companies must continually update and improve their products or risk losing market share to their competitors. Technology companies increasingly seek to ensure survival by making their products essential to consumers' technology ecosystems: Once a consumer finds a certain technology irreplaceable, she is much more likely to purchase additional electronics that are compatible with the irreplaceable product, generating an ecosystem of connected devices. The companies that own the primary devices in such systems control the entire system. These companies can limit access to their own products—creating a closed system—or they can allow anyone to create a device that fully interacts with other devices in the system, creating an open system.

This Note explores the current regulatory framework for open and closed networked systems. It recommends several reforms to curb anticompetitive conduct and retain intellectual property protection, and encourages continued innovation within an industry that depends on iterative improvements of its products. In particular, this Note focuses on the promotion of generative systems that promise the greatest possibility of innovation in the future. The goal of these recommendations is to allow small start-up companies with limited legal resources—the standard form of entry into high tech industry—to innovate without anticompetitive legal threats from established players that can spend lavishly on legal resources.

Networked systems create an incredible regulatory challenge. Inventors of such systems require incentive to create, but granting them too much control over a system's underlying ideas will stymie new inventions that the system inspires. Hence, there is a tension between strong intellectual property rights in invention and promotion of growth through iterative improvement.

Classifying systems on an open-to-closed scale measures this tension. Open systems allow broad compatibility between many different devices. Closed systems, on the other hand, tend to limit access between parts through specific rules or only allow preapproved components to connect to the system in the first place. As further detailed in Part I, each type of system exhibits its own benefits and drawbacks.

In an ideal world, regulators would seek open and stable systems, which provide the greatest potential for innovation while maintaining security, privacy, and stability. In the real world, there is a trade-off between allowing openness (or maximum compatibility and customizability) and providing security, stability, and protection for rights-holders, which in its own right spurs more innovation.

This Note focuses on two directly competing open and closed systems within the same market—mobile operating systems dominated by Google's Android and Apple's iOS—as proxies for networked systems in general. It examines each system's current regulatory field and proposes changes to antitrust and patent laws that will encourage open and generative systems, which are likely to promote the most continued innovation in the industry.

Part I examines the history of open and closed computer systems from their inception and provides brief histories of both Android and iOS. Part II describes the current regulatory structure in place for Android and iOS in terms of software and cellular phone markets under antitrust, intellectual property, and consumer protection regimes. Part III makes four recommendations to improve the competitiveness and innovative power of these markets: (1) Modernize the antitrust sham litigation claim; (2) allow reverse engineering of patents; (3) provide an independent invention defense to patent infringement; and (4) allow the use of intent evidence to analyze business improvement defenses. All of these reforms would incentivize innovation by removing legal barriers that allow established firms to enforce monopolies against start-up operations.

I. Systems Background: iOS and Android

The history and development of computer systems highlights the struggle between open accessible systems and closed compatible ones. A system is "a group of devices or artificial objects or an organization forming a network esp[ecially] for distributing something or serving a common purpose."[1] Systems combine components that may have little value on their own but substantial value when combined with complementary components. For example, a computer requires a processor, memory, storage, and an interface to be useful. By themselves, each of these components has little value: A computer processor has no value if it has no access to data, but together these components form a powerful machine capable of almost infinite uses.

In order for a system to function properly, all of its components must work together. Systems that are more open allow interoperability and portability between the different components. Closed systems have strict requirements that allow only the use of a limited set of components. Drawing a line between open and closed systems is an impossible task because one measures the level of openness relative to all other systems. Instead, the open-closed system classification runs along a scale from the theoretical maximums of completely open to completely closed.

No system can be completely open; there must be some limits in order to ensure that the individual parts work together. Likewise, no system is completely closed. For example, every computer system needs electricity to function, which means it must have a component that is compatible with an electricity delivery system. Open systems are generally compatible with a greater variety of components than closed systems. As a result, open systems allow more modification via external devices than closed ones.

Computer systems developed along two different business models: one that was almost completely closed[2] and another that was almost

[1] Webster's New Collegiate Dictionary 1184 (1973).

[2] IBM marketed the first computer mainframes, which they sold as almost completely closed systems. *See* Kevin Maney, The Maverick and His Machine 100 (2003). Starting with a punch card machine invented by Herman Hollerith, IBM primarily sold computer mainframes to the U.S. government and many of the world's largest companies by the 1960s. Jonathan Zittrain, The Future of the Internet and How to Stop It 11–12 (2008). The company's business model revolved around leasing a complete system to each client. *Id.* The leases covered hardware, software, maintenance, and training. *Id.* at 12. Each mainframe came installed with software tailored to the client. *Id.* Accordingly, IBM gave each client a customized machine for its business that could not be improved or changed without consulting IBM. *Id.* IBM designed these systems as completely closed. End users could not modify their functionality or add additional devices; only IBM software and hardware could connect to these systems. *Id.* IBM adopted this model in part because its technology was so new customers did not have the expertise or even ability to functionally modify any mainframes they purchased. *Id.* The IBM leasing model is the closest computers have ever come to a closed system.

completely open.[3] Modern computers have settled somewhere in the middle.[4]

This Note compares two mobile phone operating systems as proxies for computer systems generally: Google's Android and Apple's iOS. They are the two leading operating systems for cellular phones.[5] Each is the center of a competing ecosystem of cellular phones. As direct competitors, both systems have the same core functionality but rely on completely different business models. This makes them ideal for an analysis and comparison of the regulatory framework for high tech networked systems.

Apple's iOS is based on the same philosophy as the 1984 Macintosh and the original IBM punch card machines from the 1890s.[6] iOS is only compatible with one set of hardware: the iPhone. When it was released in 2007, no one had ever seen a system like it. iOS is a mostly closed system:[7] It is only compatible with Apple hardware[8] and can only connect to computers through Apple-made software.[9] Any semblance of openness in iOS originates in its App Store, which allows third-parties to interface between their products and iOS. Even so, Apple only allows users to install applications through its carefully controlled iTunes service, and it

[3] The first personal computers exhibited the characteristics of open systems. In their infancy, companies like Intel and Texas Instruments sold individual components, like processors and memory chips, that hobbyists combined together to create the first personal computers. *See* PAUL E. CERUZZI, A HISTORY OF MODERN COMPUTING 222–26 (I. Bernard Cohen & William Aspray eds., 2d ed. 2003); ZITTRAIN, *supra* note 2, at 13. These systems were completely open—the hobbyist purchased each individual component separately and assembled the machine herself. For example, some of the earliest systems could use standard televisions as displays and cassette players to store data. *Id.* The user could install the software of her choice on her system. *Id.* In the early years, personal computers were mostly a hobbyist's activity that required interchangeability between parts. Consequently, open design choices reigned.

[4] As computer systems became more accessible, their design gravitated toward the center of the spectrum, although not completely by choice. Entrepreneurs like Steve Jobs understood that mass-producing a standardized personal computer complete with all necessary components meant a more reliable, efficient, and cheaper computer for consumers. CERUZZI, *supra* note 3, at 264. The company he co-founded, Apple, Inc., released the forerunner to modern computer systems, the Macintosh, in 1984. *Id.* at 273. Unlike previous personal computers, the Macintosh was closed, meaning users could not add additional hardware or modify the physical system itself. *Id.* at 275. Additionally, because Apple wanted to retain control over the entire system, and for technical reasons, users were required to use Apple's own operating system. *Id.* at 276. Despite this restriction, users could install whatever programs they wished on to their Apple computers without Apple's permission. *See* ZITTRAIN, *supra* note 2, at 16. In fact, Apple encouraged users to write programs that could run on the operating system, so the entire system remained relatively open. *See id.* at 11–18.

[5] Zack Epstein, *Android and iOS Regain Market Share in January as BlackBerry Slides*, BGR, (Feb. 1, 2012, 3:40 PM), http://www.bgr.com/2012/02/01/android-and-ios-regain-market-share-in-january-as-blackberry-slides.

[6] *See supra* note 2; CERUZZI, *supra* note 3, at 275.

[7] The original iOS offered no App Store and disallowed users from installing any of their own software on the phone itself.

[8] Compatible hardware includes the iPhone and, more recently, the iPod Touch and the iPad. *See* APPLE IOS, http://www.apple.com/ios (last visited July 30, 2013).

[9] *See* APPLE ITUNES, http://www.apple.com/itunes (last visited July 30, 2013).

carefully vets applications before they reach the market.[10] Furthermore, Apple restricts programmers from accessing the full power of the iPhone's hardware itself.[11]

The development of Google's Android system is analogous to the personal computer revolution.[12] Since the beginning, it has been an open-source system: Any phone manufacturer can make a phone that runs on Android. Google does not restrict applications from running on its system.[13] Furthermore, development is not artificially restricted[14] as on Apple's platform; software on the Android system can access the same interfaces available to its manufacturers.

Both iOS and Android contain the same core functionality. They are direct substitutes in an economic sense and face similar regulatory problems from a competition and innovation perspective. However, one developed as a closed system and the other as an open one. A review of the existing regulatory body of law applicable to both systems underscores the significance of this distinction.

II. Overview of the Existing Regulation of Android and iOS

Existing regulation of Android and iOS falls into two categories: antitrust and intellectual property. Antitrust regulation seeks to ensure a competitive marketplace and to prevent anticompetitive monopolization of a market. Intellectual property—in this case patents—rewards inventors by granting them a limited monopoly over the use of their inventions for a limited period of time.

A. Antitrust

Congress designed American antitrust law to protect competition and consumers. Antitrust law encompasses two main areas: combinations in restraint of trade (violations of section 1 of the Sherman Act)[15] and

[10] *See* Apple App Review, https://developer.apple.com/appstore/guidelines.html (last visited July 30, 2013).

[11] *See* Apple iOS App Programming Guide, https://developer.apple.com/library/ios/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html (last visited July 30, 2013).

[12] *See supra* note 3.

[13] Google does not subject applications to pre-screening before allowing them onto its online marketplace, but reserves the right to remove them if they violate the terms of service. In any event, the Android software itself allows users to install programs directly onto the phone, a feature that iOS lacks. Google's app distribution system "Google Play" is optional for developers. They can distribute apps using their own sources. *See* Android Other Developer Servs., http://developer.android.com/legal.html (last visited July 30, 2013). For an example of third-party app distribution systems, see Simon Hill, *Tired of Google Play? Check out these Alternative Android App Stores*, Digital Trends (Feb. 4, 2013), http://www.digitaltrends.com/mobile/android-app-stores.

[14] *See supra* note 2.

[15] 15 U.S.C. § 1 (2012).

monopolization of trade (violations of section 2 of the Sherman Act).[16] Section 1 violations occur when two or more parties conspire to restrain trade through an agreement.[17] A company violates section 2 when it possesses or attempts to possess monopoly power in the relevant market and willfully maintains that power in an anticompetitive manner, "distinguished from growth or development as a consequence of a superior product, business acumen, or historic accident."[18] Monopolization generally requires a showing that a company attempted, acquired, or maintained market power in a relevant market defined by product and geography through exclusionary conduct.[19]

Courts have struggled defining both the relevant market and exclusionary conduct. This Note does not discuss problems related to relevant market definition because it focuses on regulation of a specific market. Both iOS and Android possess large market shares and the ability to affect both the price and market output for cell phones and operating system software. In other words, each has monopoly power[20] in the market. As a result, any action taken by either company will have a dramatic effect on those markets as a whole. The discussion below focuses on these markets intricacies. However, monopoly power is required to sustain claims of tying, predatory innovation, or refusal to deal or license. These antitrust concerns will therefore only apply to market-dominating products like iOS and Android, as opposed to all networked systems.

The difficulty in applying antitrust principles to new technology lies in defining exclusionary, or anticompetitive, conduct. Courts have long been wary of applying the Sherman Act to markets in which they have little experience because they have been reluctant to over-regulate the economy.[21] The primary types of exclusionary conduct involving technology include tying,[22] predatory innovation,[23] refusal to deal or license in the

[16] *Id.* § 2.

[17] *Id.* § 1 ("Every contract, combination in the form of a trust or otherwise, or conspiracy, in restraint of trade . . . ."). This Note concerns unilateral actions of competitor companies who do not license to each other and who seem intent on competition with each other so it will not focus on section 1 violations. *See* Erica Ogg, *Steve Jobs Vowed to "Destroy" Android*, GIGAOM (Oct. 21, 2011, 6:56 AM), http://www.gigaom.com/2011/10/21/steve-jobs-vowed-to-destroy-android.

[18] United States v. Grinnell Corp., 384 U.S. 563, 571 (1966).

[19] United States v. Microsoft Corp., 253 F.3d 34, 58 (D.C. Cir. 2001).

[20] The Supreme Court has alternatively defined monopoly power as "the ability to raise prices above those that would be charged in a competitive market," *NCAA v. Bd. of Regents of Univ. of Okla.*, 468 U.S. 85, 109 n.38 (1984), or "the power to control prices or exclude competition." United States v. E. I. du Pont de Nemours & Co., 351 U.S. 377, 391 (1956). For a more thorough discussion of the definition of monopoly power (a term used synonymously with market power) *see* Thomas G. Krattenmaker et al., *Monopoly Power and Market Power in Antitrust Law*, 76 GEO. L.J. 241 (1987).

[21] *See* Broad. Music, Inc. v. Columbia Broad. Sys., Inc., 441 U.S. 1, 9–10 (1979).

[22] *See* Eastman Kodak Co., v. Image Technical Serv., Inc., 504 U.S. 451, 461–62 (1992).

patent context,[24] and sham intellectual property infringement actions.[25] This Note considers each type of conduct in the context of Android and iOS and the economic activities of smaller companies that rely on these systems as the bases for their innovation.

1. Tying

"A tying arrangement is 'an agreement by a party to sell one product but only on the condition that the buyer also purchases a different (or tied) product, or at least agrees that he will not purchase that product from any other supplier.'"[26] A tying agreement is unlawful where it substantially impacts interstate commerce, where the seller offers two products exclusively as a unit, and where the seller has "economic power" in the tying product.[27] Tying arrangements are anticompetitive because they permit the seller to exploit its "control over the tying product to force the buyer into the purchase of a tied product that the buyer either did not want at all, or might have preferred to purchase elsewhere on different terms."[28]

In a seminal tying case, *Eastman Kodak Co. v. Image Technical Services* ("*Kodak I*"), the Supreme Court examined Kodak's replacement parts policy and articulated tying doctrine as applied to a technologically innovative market.[29] When Kodak began selling high-volume photocopiers, it allowed independent services organizations ("ISOs") to buy replacement parts directly from the company or the original licensed manufacturer.[30] Kodak later amended its policy and only sold replacement parts to buyers of Kodak equipment or end users to repair their own equipment.[31] As a result, many ISOs found themselves unable to obtain parts and Kodak forced them out of the market for servicing Kodak machines.[32] Image Technical Services, Inc. alleged that Kodak's amended parts policy violated section 2 because it tied the market for services to the market for parts.[33]

Kodak I had two significant holdings. First, a single brand of a product or service can be a relevant market under the Sherman Act.[34]

[23] *See* Berkey Photo, Inc. v. Eastman Kodak Co., 603 F.2d 263, 281 (2d. Cir 1979); Foremost Pro Color, Inc. v. Eastman Kodak Co., 703 F.2d.534, 544–45 (9th Cir. 1983).

[24] *See* Image Technical Serv., Inc. v. Eastman Kodak Co., 125 F.3d 1195, 1216–17 (9th Cir. 1997); Data Gen. Corp. v. Grumman Sys. Support Corp., 36 F.3d 1147, 1186 (1st Cir. 1994).

[25] *See* In re Indep. Serv. Orgs. Antitrust Litig, 203 F.3d 1322, 1326 (Fed. Cir. 2000).

[26] *Eastman Kodak Co.*, 504 U.S. at 461 (quoting N. Pac. Ry. Co. v. United States, 356 U.S. 1, 5–6 (1958)).

[27] Jefferson Parish Hosp. Dist. No. 2 v. Hyde, 466 U.S. 2, 12 (1984).

[28] *Id.* at 12.

[29] *Eastman Kodak Co.*, 504 U.S. at 457.

[30] *Id.* at 458.

[31] *Id.*

[32] *Id.*

[33] *Id.* at 459.

[34] *Id.* at 481–82.

This is significant because it opens the door for suits against companies that attempt to control industries that stem from a product they have created. Second, the Court held that as a matter of law, a "valid business justification" can excuse actions that would otherwise be inconsistent with section 2 of the Sherman Act.[35] Valid business justifications include, among other things, technical improvements to products or services including increased compatibility and security. More recently, courts have been reluctant to dismiss justifications provided by manufacturers as invalid business justifications for fear of stifling innovation.[36]

It is theoretically possible to frame a tying case against either Android or iOS. In the case of iOS as the tying product, the tied product could be either the iPhone itself,[37] or it could be the iTunes App Store, which serves as the exclusive gateway for customers to obtain digital content on their phones.[38] Apple certainly has "economic power" in the tying product: It is the sole producer of iOS, just as Kodak is the sole producer of its copy machines. Sales of iPhones and iOS certainly affect a substantial amount of interstate commerce. An analogous prima facie claim would replace Android as the tying product and the Android Market as the tied product.[39]

A theoretical tying claim regarding Android or iOS would not and should not survive in court. Both Apple and Google have many valid business justifications, as outlined in *Kodak I*. Most prominent among these is the need for compatibility, security, and simplicity for consumers to use their devices. Given the reluctance of courts to declare business justifications invalid, these justifications can and should prevent plaintiffs from suing cell phone operating system manufacturers for tying under section 2.

The Court's unwillingness to find a tying violation in light of a valid business justification creates a large loophole for platform owners, such as Apple and Google, to use against smaller entities looking to build upon those systems. For example, Apple has prevented the installation of certain applications on iOS that may compete with an application made

[35] *Id.* at 483.

[36] *See* United States v. Microsoft Corp., 253 F.3d 34, 65 (D.C. Cir. 2001).

[37] Apple considers the iPhone to be part of the iOS system and therefore part of the market. *See* Apple iOS, http://www.apple.com/iphone/ios (last visited July 30, 2013). Two rationales undercut this argument. First, Android, iOS's strongest competitor, performs the same core functions but is designed independently of the phone's hardware. Second, iOS is installable on two different devices, one of which is not a cellular phone.

[38] *See* Jeffrey Jarosch, *Novel "Neutrality" Claims Against Internet Platforms: A Reasonable Framework for Initial Scrutiny*, 59 Clev. St. L. Rev. 537, 584 (2011).

[39] Note that the more open Android system is less likely to lead to illegal tying because it promotes compatibility between itself and any other component, while the closed iOS system can theoretically be tied to any outside system with which it interacts.

by Apple.[40] In essence, Apple could block any software that it feels competes with one of the core features of iOS, essentially tying iOS and what could be a proprietary feature together. A business justification is easy in that scenario: Apple could claim that a version of the software is more secure and prevents hardware failure at a better rate than its competitor's version. This is certainly a valid business justification under the current framework. Conveniently, it also allows Apple to eliminate competitors—especially small firms that rely on Apple's iOS as a platform to distribute their products—and potential innovators from the market.[41]

On the other hand, open systems like Android promote compatibility and interoperability between the maximum number of components. To this end, Android does not restrict installation of specific applications through the Play Store, its version of the App Store. Further, it allows users to install applications from sources outside the Play Store. This exemplifies the competitive benefits of open systems.

2. *Predatory Innovation*

A predatory innovation claim alleges that that a monopolist changed its product specifically to interfere with competitors.[42] For example, a plaintiff could allege that a defendant company changed compatibility specifications for its product solely to disqualify a competitor's product from connecting to it. This type of claim is incredibly difficult to prove.[43] Such a claim under section 2 infers that a "monopolist, no less than any other competitor, is permitted and indeed encouraged to compete aggressively on the merits, and any success it may achieve solely through 'the process of invention and innovation' is necessarily tolerated by the antitrust laws."[44] Courts have rightly been concerned with predatory innovation claims for two reasons. First, allowing courts to oversee product design "would be contrary to the very purpose of the antitrust laws, which is, after all, to foster and ensure competition on the merits."[45] Second,

[40] In 2009, for example, Apple blocked a Google Voice application from installation on iOS because it "duplicates features that come with the iPhone." Erica Ogg, *Apple Blocks Google Voice App for iPhone*, CNET (July 28, 2009, 1:25 PM), http://news.cnet.com/8301-13579_3-10297618-37.html.

[41] Google can theoretically take similar anticompetitive actions but, due to its commitment to Android as an open system and lack of restriction on the source of applications (consumers can install applications on Android without interfacing with Google's online application repository), it will be much less successful if it tries to do so.

[42] Allied Orthopedic Appliances, Inc. v. Tyco Health Care Grp. LP, 592 F.3d 991, 999 (9th Cir. 2010).

[43] *See id.* at 998–99 (explaining that a design change by itself does not violate section 2 even if it is performed by a monopolist and harms competitors as a result); United States v. Microsoft Corp., 253 F.3d 34, 65 (D.C. Cir. 2001) ("As a general rule, courts are properly very skeptical about claims that competition has been harmed by a dominant firm's product design changes.").

[44] Foremost Pro Color, Inc. v. Eastman Kodak Co., 703 F.2d 534, 544–45 (9th Cir. 1983) (quoting Berkey Photo, Inc. v. Eastman Kodak Co., 603 F.2d 263, 281 (2d Cir.1979)).

[45] *Id.* at 544.

any attempt by the courts to weigh the benefits of design improvements and their anticompetitive effects would not be administrable.[46]

Predatory innovation claims have been successful even though they face an uphill battle. In *United States v. Microsoft*, the D.C. Circuit held that Microsoft harmed competition by integrating Internet Explorer into the Windows 98 Operating System without showing "that its conduct serve[d] a purpose other than protecting its operating system monopoly."[47] In *Microsoft*, the court found that although Microsoft made a general claim that its vision for "deeper levels of technical integration is highly efficient and provides substantial benefits to customers and developers[,]" it did not specify or substantiate those claims.[48] The court found that Microsoft failed to meet its burden to show that its conduct served a purpose other than protecting its monopoly.[49] Under *Microsoft*, any design improvement justifies an unassailable defense to antitrust liability under section 2. However, that improvement must be identified in sufficient detail in order for a court to identify it as such.

It is possible to conceive of predatory innovation claims against Android and iOS. In fact, Apple has defeated similar claims in the past. In *In re Apple iPod iTunes Antitrust Litigation*, the plaintiff challenged two design changes to Apple's iPod software.[50] These changes eliminated compatibility between iPods and third-party software and changed the encryption method which Apple used to transfer digital content to iPods.[51] These changes rendered the plaintiff's products incompatible with Apple iPods.[52] Even with a public statement in evidence stating that Apple knew that its changes might break compatibility with the plaintiff's products[53] and Apple's continued refusal to license its new changes to the plaintiff,[54] the court granted summary judgment in Apple's favor on one claim and reserved the other for further factual findings.[55] The court determined that this valid business justification was enough to evade antitrust liability.

Notice the structure of the above predatory innovation claim: It devolves from an action that takes an open system—the Apple iPod—

[46] *Allied Orthopedic*, 592 F.3d at 1000 ("There are no criteria that courts can use to calculate the 'right' amount of innovation, which would maximize social gains and minimize competitive injury. A seemingly minor technological improvement today can lead to much greater advances in the future. The balancing test proposed by plaintiffs would therefore require courts to weigh as-yet-unknown benefits against current competitive injuries.").

[47] United States v. Microsoft Corp., 253 F.3d 34, 67 (D.C. Cir. 2001).

[48] *Id.* at 66 (internal quotation marks omitted).

[49] *Id.* at 67.

[50] 796 F. Supp. 2d 1137, 1143 (N.D. Cal. 2011).

[51] *Id.* at 1143, 1146.

[52] *Id.*

[53] *Id.* at 1144.

[54] *Id.* at 1145.

[55] *Id.* at 1148.

and closes it.[56] By definition, this change eliminates compatibility among formerly compatible components. This could lead automatically to a predatory innovation claim if there were no pro-competitive justification for the actions limiting the compatibility with different devices. Because procompetitive justifications are easy to come by,[57] it is simple for a technology company to improve its products that cut off compatibility with all competitive or reliant products. These changes help established software companies control markets that they have created but completely stifle innovation from startup competitors, essentially foreclosing competition and perhaps stagnating innovation in the market.

Open systems are much less likely to face predatory innovation claims because they rarely limit compatibility with other components. For example, Android would not likely face a claim like the one against Apple because as an open system, it does not limit compatibility to certain types of software or hardware other than those that meet minimum industry standard specifications. Similarly, open systems are much less likely to change their parameters to foreclose competition without a significant reason for the change because open systems thrive on the potential for independent developers to add compatible innovative components.

Inventors of widely used technology systems are most likely to use predatory innovation to capture as much of the market they created as possible. In the examples above, Microsoft and Apple sought to foreclose competitors from providing a service directly related to a system that they created. Proprietors of closed systems are by definition much more likely to engage in such activity, because such systems restrict interoperability and compatibility more than open ones.

3. *Refusal to Deal or License*

Under the Sherman Act, a company "generally has a right to deal, or refuse to deal, with whomever it likes, as long as it does so independently."[58] But when a company has an established policy of dealing with competitors and subsequently refuses to deal with them without any efficiency justification, it can run afoul of competition laws.[59] Like predatory innovation, cessation of dealings without any efficiency justification is anticompetitive because the refusing company likely hopes that its cessation of dealings will hurt its competitor more than itself and that it can capture any market share lost by its competitors.

[56] The iPod was an open system because many different pieces of software could read and write to it. By changing the iPod so that only iTunes could communicate with it, Apple closed the iPod system.

[57] Examples of procompetitive justifications include security, ease of use, or protection of copyrighted material.

[58] Monsanto Co. v. Spray-Rite Serv. Corp., 465 U.S. 752, 761 (1984) (citing United States v. Colgate & Co., 250 U.S. 300, 307 (1919)).

[59] *See* Aspen Skiing Co. v. Aspen Highlands Skiing Corp., 472 U.S. 585, 603–08 (1985).

Intellectual property rights significantly complicate refusal to deal claims in the realm of technology. When a company holds a patent on a product, it has the right to exclude sales of that product to any other party and to choose whether to license the patent at all.[60] No "court has imposed antitrust liability for a unilateral refusal to sell or license a patent or copyright."[61] This protection under patent law is significant in any market involving innovative technology because all competitors are likely to maintain large patent portfolios to guard their rights.[62] These protections grant technology companies—like Google and Apple—significant leeway to refuse to deal with anyone. Given that these exclusionary rights last for twenty years,[63] an eternity in technology markets, patent protection in effect forecloses all competition or use of the patent subject matter for a period long enough to ruin any startup effort to innovate in that field.

Courts have approached patent-related refusal to deal or license in several different ways.[64] In *In re Independent Service Organizations Antitrust Litigation* ("*Xerox*"), the Federal Circuit emphasized the right to exclude granted by the patent system, but it carved out "three limited categories in which a patent holder would not be immune from antitrust liability: (1) tying patented and unpatented products; (2) obtaining a patent through knowing and willful fraud; and (3) engaging in sham litigation."[65] Any action within the scope of a patent grant cannot violate antitrust laws, and the court would not examine the subjective intent of the refusal to deal with the competitor.[66]

The Ninth Circuit applied antitrust law more expansively in *Image Technical Services, Inc. v. Eastman Kodak Co.* ("*Kodak II*").[67] Instead of granting holders carte blanche to exclude or refuse to deal or license, the court held that the use of a patent to exclude others is a presumptively

[60] *See* United States v. Westinghouse Elec. Corp., 648 F.2d 642, 646–47 (9th Cir. 1981) (quoting Simpson v. Union Oil Co., 377 U.S. 13, 24 (1964)) ("The patent laws which give a 17-year monopoly on 'making, using or selling the invention' are in *pari materia* with the antitrust laws and modify them *pro tanto*.").

[61] Image Technical Servs., Inc. v. Eastman Kodak Co., 125 F.3d 1195, 1216 (9th Cir. 1997). The court also noted that it makes no difference whether a "case involves a selective refusal to sell products protected by patents and copyrights, [and] not an absolute refusal to license." *Id.* at 1216 n.9.

[62] *See, e.g.*, Evelyn M. Rusli & Claire Cain Miller, *Google to Buy Motorola Mobility for $12.5 Billion*, N.Y. TIMES (Aug. 15, 2011, 7:34 AM), http://dealbook.nytimes.com/2011/08/15/google-to-buy-motorola-mobility ("Our acquisition of Motorola will increase competition by strengthening Google's patent portfolio, which will enable us to better protect Android from anticompetitive threats from Microsoft, Apple and other companies . . . .").

[63] 35 U.S.C. § 154 (2012).

[64] For a complete discussion of the courts approach to these types of cases, *see* Michael A. Carrier, *Unraveling the Patent-Antitrust Paradox*, 150 U. PA. L. REV. 761 (2002).

[65] *See* Carrier, *supra* note 64, at 776–77 (citing In re Indep. Serv. Orgs. Antitrust Litig., 203 F.3d 1322, 1326 (Fed. Cir. 2000)).

[66] *In re* Indep. Serv. Orgs. Antitrust Litig., 203 F.3d 1322, 1327–28 (Fed. Cir. 2000).

[67] 125 F.3d 1195 (9th Cir. 1997).

valid business justification.[68] Upon remand from the Supreme Court, the Ninth Circuit rejected Kodak's argument that patents on the parts it now refused to sell permitted them to refuse to deal without conflicting with the Sherman Act.[69] The court held that Kodak could not use patent protections as a "pretext" to refuse to deal with or sell to competitors.[70] The Ninth Circuit went further to hold that evidence of the subjective intent and state of mind of Kodak employees can show pretext.[71] Given that a manager at Kodak "testified that patents 'did not cross [his] mind' at the time Kodak began the parts policy," the court found that Kodak could not use patent grants as a post-hoc rationalization for amending its policy to refuse to deal with ISOs.[72]

The principle difference between the Ninth and Federal Circuit approaches lies in the examination of subjective intent evidence. The Federal Circuit refused to consider such evidence because the patent holder in *Xerox* merely enforced its statutory rights.[73] Without evidence that the anticompetitive effect of the refusal to deal extended beyond the statutory grant, the court would not consider evidence of intent.[74]

The Ninth Circuit's approach seems more equitable than the Federal Circuit's. While the goal of avoiding what may turn out to be a complex task of establishing intent is admirable, it is fundamentally unfair for a patent holder to hide behind the protections of that patent[75] as a post hoc rationalization. This is especially the case when a defendant firm has licensed the patent for use to the complaining entity in the past, only to withdraw licensing for an undisclosed reason, maintaining the defense: "We didn't want to license anymore and we have a patent." Declaring a defense purely under the scope of the patent smacks of anticompetitive conduct and clearly extends beyond the public policy supporting exclusive control of patented subject matter by a patent holder.[76]

[68] *Id.* at 1218.

[69] *Id.* at 1219.

[70] *Id.* ("Neither the aims of intellectual property law, nor the antitrust laws justify allowing a monopolist to rely upon a pretextual business justification to mask anticompetitive conduct."). Pretext can include acquisition of patents through fraud as in *Xerox. Id.*

[71] *Id.*

[72] *Id*. (alteration in original). *But see In re* Indep. Serv. Orgs. Antitrust Litig., 203 F.3d 1322, 1327–28 (Fed. Cir. 2000) (finding that right to exclude granted by patent is not modified by intent, therefore intent is irrelevant in patent-antitrust determination).

[73] *In re Indep. Serv. Orgs.*, 203 F.3d at 1327.

[74] *Id.* at 1328.

[75] On such protection afforded to a patent holder is a limited monopoly over the invention, including the right of exclusion.

[76] Two of the main policies underlying the right of exclusion to patent-holders are to encourage new inventions by guaranteeing the subsequent right of control and to place inventions in the public domain. When a firm has a patent over an invention and licenses that invention, it is taking advantage of

Refusal to deal in the context of large platforms like operating systems is problematic for innovation in the software industry. Small firms rely on the widespread availability and market penetration of these platforms when they release technology. Allowing platform owners to refuse to deal for any valid business justification permits them to cannibalize the business of smaller firms who have improved that platform by releasing products within it. For example, it would be entirely possible for Google to decide that it wants to modify Android to take on a degree of functionality that is already available on a popular third-party application. Google could refuse to allow users to install that application on Android, which would completely and anti-competitively preempt the firm that created the application. In this scenario, Google would have complete immunity because it could rely on any number of valid business justifications, such as increased security and stability of the platform itself. This justification should not enable large platform developers to co-opt market share from firms that justifiably rely on them to disseminate their products.[77]

It remains clear from the examples above that a company has no duty to deal with any of its competitors absent previous dealings because companies have the right to choose with whom they deal.[78] Apple has never explicitly licensed or sold iOS to anyone else and refuses to do so.[79] Therefore, Apple remains impervious to refusal to deal suits as it has never dealt with competitors in the past. By extrapolation, closed systems are less likely to deal with competitors because they lack interoperability and compatibility with many outside devices. Therefore, they are less likely to be liable for refusal to deal under current antitrust laws.

Google, on the other hand, has a blanket policy of licensing Android to anyone.[80] An open system like Android, run by a large corporation like Google, frequently deals with competitors. If it were to revoke those licensing agreements without a business justification, it could face antitrust scrutiny.[81] Google would preclude liability, however, if it had a genuine

its patent right of control. A subsequent refusal to deal on the grounds that it has the right to do so under the patent is anticompetitive and partially removes the patented material from the public domain.

[77] Recently, Apple committed a lighter version of this when it replaced Google Maps with its own proprietary mapping software. However, Apple did not restrict Google from creating its own separate mapping application. Even so, Apple Maps continues to act as the permanent default on iOS, restricting some of the functionality once available to Google Maps.

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

[79] Apple has offered to license patents underlying iOS but has not yet completed any public deals. Nilay Patel, *Apple Licensed iOS Scrolling Patent to Nokia and IBM, Offered License to Samsung*, THE VERGE (Dec. 3, 2011, 4:28 PM), http://www.theverge.com/2011/12/3/2608407/apple-license-ios-scrolling-patent-nokia-ibm-offered-samsung.

[80] *Licenses*, ANDROID OPEN SOURCE PROJECT, (last visited July 30, 2013) http://source.android.com/source/licenses.html.

[81] Google's recent acquisition of Motorola Mobility Ltd., an actual manufacturer of cell phones, raised these concerns among the Department of Justice and competitors. *See* Rusli & Miller, *supra* note 62.

business reason for refusing to license. Overall, by their nature, open systems are more likely to face antitrust liability for refusal to deal, but they are only likely to do so if they exclude former partners from access to the system, which would move them toward the closed end of the spectrum.

As discussed above, inventors of both open and closed systems face a litany of antitrust concerns. Closed systems are more likely to face claims of illegal tying and predatory innovation. Open systems are more likely to face refusal to deal claims, but only if they transition towards a more closed environment.

4. *Antitrust Concerns Surrounding Intellectual Property Infringement Actions and Sham Litigation*

However, use of the court system through sham patent infringement litigation or by misuse of a patent in a monopolization scheme can give rise to antitrust liability.[82] In *Mercoid Corp. v. Mid-Continent Investment Co.*, the Supreme Court found a valid antitrust cause of action where a company "conspired to establish a monopoly in an unpatented appliance beyond the scope of the patent and in violation of the anti-trust laws."[83] The Court expanded this cause of action in *Walker Process Equipment Inc. v. Food Machinery*, where it held that "the enforcement of a patent procured by fraud on the Patent Office may be violative of § 2 of the Sherman Act."[84] A *Walker Process* claim requires showing that a patent holder is attempting to enforce a fraudulently held patent in order to monopolize a market.[85] Lower courts have further fleshed out these claims to allow antitrust claims based on illegitimate patent usage.[86]

Fraud as perpetuated in *Mercoid* and *Walker Process* is a subset of an exception to *Noerr-Pennington* immunity[87] to petition the government to take anticompetitive action against one's competitors.[88] The First Amendment protects the ability of all persons to petition the government.[89] However, when a monopolist petitions the government to take action that would provide an anticompetitive benefit in its favor,

[82] *See* Mercoid Corp. v. Mid-Continent Inv. Co., 320 U.S. 661, 664–65 (1944).

[83] *Id.* at 662.

[84] 382 U.S. 172, 174 (1965).

[85] *Id.* at 175.

[86] *See, e.g.*, Kobe, Inc. v. Dempsey Pump Co., 198 F.2d 416 (10th Cir. 1952) (holding that pooling of patents to maintain monopoly gives rise to antitrust action); Handgards, Inc. v. Ethicon, Inc., 601 F.2d 986, 993 (9th Cir. 1979) (stating that enforcement of a knowingly invalid patent can give rise to antitrust liability).

[87] *See generally* E. R.R. Presidents Conference v. Noerr Motor Freight, Inc., 365 U.S. 127, 135 (1961); United Mine Workers v. Pennington, 381 U.S. 657 (1965).

[88] *See* S.W. O'Donnell, *Unified Theory of Antitrust Counterclaims in Patent Litigation*, 9 VA. J.L. & TECH. 8, 27 (2004).

[89] U.S. CONST. amend. I.

First Amendment freedoms collide with competition law. As one might expect, First Amendment constitutional rights trump those delineated in the Sherman Act: "[N]o violation of the [Sherman] Act can be predicated upon mere attempts to influence the passage or enforcement of laws."[90] *Noerr-Pennington* immunity has expanded to situations including redress through judicial or quasi-judicial bodies.[91]

The primary exception to *Noerr-Pennington* immunity is the sham exception.[92] In *Professional Real Estate Investors, Inc. v. Columbia Pictures Industries, Inc.*, the Supreme Court articulated the modern standard to determine whether a lawsuit constitutes sham litigation for purposes of the Sherman Act.[93] "First, the lawsuit must be objectively baseless in the sense that no reasonable litigant could realistically expect success on the merits."[94] An objectively baseless suit is one in which the plaintiffs have no probable cause to institute the suit in the first place, similar to the probable cause standard used in the common law tort of wrongful civil proceedings.[95] If a plaintiff first shows that the underlying suit was objectively baseless, she can then move to the second element of the test: to show that the subjective purpose of the litigation is to invoke government processes in a scheme to "interfere directly with the business relationships of a competitor."[96] *Noerr-Pennington* immunity in the context of sham litigation is incredibly difficult to overcome, because the objective part of the test renders any lawsuit with a remote chance of succeeding impervious to objection under the Sherman Act.

Whether a system is more open or closed does not bear on whether it is susceptible to sham litigation antitrust violations. These violations go to the character of the system's designers. Even so, sham litigation represents a serious threat against smaller firms looking to innovate in markets related to iOS and Android. Small technology firms are usually composed of few employees, very few of whom (if any) are lawyers. The presence of a threat to their business in the form of an intimidating notice and pending lawsuit directed at the firm is incredibly likely to stop development of any product in its tracks. A small firm is unlikely to have the legal expertise to evaluate the claim and to have the resources to hire legal counsel. In the end, the firm is more likely to abandon the project in the face of intense pressure than risk losing everything in court. The litigation may have been

[90] *Noerr*, 365 U.S. at 135. For an extended discussion of the origins of *Noerr-Pennington* immunity, see Sosa v. DirecTV, Inc., 437 F.3d 923, 929–32 (9th Cir. 2006).

[91] Cal. Motor Transp. Co. v. Trucking Unlimited, 404 U.S. 508, 510–12 (1972).

[92] *See Noerr*, 365 U.S. at 144.

[93] 508 U.S. 49, 60–61 (1993).

[94] *Id.* at 60.

[95] *Id.* at 62.

[96] *Id.* at 60–61 (emphasis omitted) (quoting *Noerr,* 365 U.S. at 144).

a sham, but the firm will not contest it to find out, especially where the burden for proving the sham is stacked against the small firm.

B. Patent

In addition to antitrust concerns, and of more importance to small companies, developers of networked systems rely on patent protections to incentivize investment in their products. Both Apple and Google use patents to protect their inventions. The Constitution grants Congress the power to "promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."[97] Congress responded by passing the Patent Act of 1790, establishing the patent system still in place in the United States today.[98] An inventor of "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement therefor" may patent that invention.[99] This broad definition of patentable subject matter makes almost anything patentable, including pharmaceuticals, chemical processes, and more controversially, software and business methods.[100] Furthermore, patentable items must be useful, novel, and nonobvious.[101] For prospective patent holders, the novelty and non-obvious requirements present the biggest hurdle.

Approved patents provide their inventors control over the patentable subject matter.[102] Patent holders can recover damages, attorney fees, and even receive an injunction that prevents others from using the patented subject matter.[103] These are significant powers that allow inventors to prevent others from copying their work.[104] The Federal Circuit has exclusive jurisdiction to hear appeals on all patent cases.[105] This Subpart discusses the effects of enforcement of patent rights on open and closed systems. Developers of closed systems can and do leverage the exclusive use rights granted in the patent monopoly as a tool to prevent others from accessing or competing with their systems.

[97] U.S. Const. art. I § 8., cl. 8.

[98] *See* 35 U.S.C. §§ 1–376 (2012).

[99] *Id.* § 101.

[100] *See generally* Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Calif. L. Rev. 1 (2001); John F. Duffy, *Why Business Method Patents?*, 63 Stan. L. Rev. 1247 (2011).

[101] *See* 35 U.S.C §§ 101–03.

[102] *See id.* § 271 ("[W]hoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent.").

[103] *See id.* §§ 283–85.

[104] Most importantly, patent law contains no fair use or reverse-engineering exemptions. *See generally* Cohen & Lemley, *supra* note 100.

[105] 28 U.S.C. § 1295 (2012).

1. The Particular Problem of Overbroad Software Patents

Software patents are a particular tool used by both Apple and Google to protect iOS and Android.[106] Apple and Google hold large patent portfolios to prevent others from copying their inventions. For iOS and Android, those patents protect the software. The patentability of software has been controversial ever since approval of the first software patents.[107] In 1972, the Supreme Court rejected a patent for a software method applicable to a general purpose computer of any type.[108] However, the Court soon changed its mind following the computer industry's continued growth. In 1981, the Court upheld a patent that included a mathematical formula run by software in conjunction with significant "postsolution activity" outside the computer program in *Diamond v. Diehr*.[109] In other words, software used in conjunction with already patentable subject matter itself became patentable. This ruling spawned "the doctrine of magic words" for which "software was patentable subject matter, but only if the applicant recited the magic words and pretended that she was patenting something else entirely."[110] The Supreme Court has assiduously refused to discuss the patentability of software since *Diehr*.[111] Later Federal Circuit decisions have abrogated even the need for "magic words."[112] The modern test for patentability is whether a process passes the "useful, concrete, and tangible result" test.[113]

The patentability of software presents several difficulties for the modern patent system and innovation in software systems like iOS and Android. First, a valid patent specification requires a description of the innovation ("enablement") and a description of the "best mode" of implementing the innovation to allow practitioners to recreate it.[114] These

[106] Apple has patents protecting hardware related to iOS as well, but those patents are not as problematic as purse software patents. *See infra* note 133.

[107] For a more complete history of the patentability of software, *see* Cohen & Lemley, *supra* note 100, at 8–11 and Andrew Nieh, Note, *Software Wars: The Patent Menace*, 55 N.Y.L. Sch. L. Rev. 295, 300–06 (2010).

[108] Gottschalk v. Benson, 409 U.S. 63, 72–73 (1972) (holding that patenting of computer algorithms alone is impermissible).

[109] 450 U.S. 175, 191–92 (1981).

[110] Cohen & Lemley, *supra* note 100, at 9.

[111] In *Bilski v. Kappos*, the Court rejected the machine-or-transformation test as the only test for determining patent eligibility of a process. 130 S. Ct. 3218, 3227 (2010). But the Court refused to reconsider prior decisions by the Federal Circuit upholding software patents. *Id.* at 3248 n.40.

[112] *See generally* State St. Bank & Trust Co. v. Signature Fin. Grp. Inc., 149 F.3d 1368 (Fed. Cir. 1998); *In re* Alappat, 33 F.3d 1526 (Fed. Cir. 1994).

[113] AT&T Corp. v. Excel Commc'ns Inc., 172 F.3d 1352, 1358 (Fed. Cir. 1999).

[114] 35 U.S.C. § 112 (2012) ("The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor or joint inventor of carrying out the invention.").

requirements stem from one of the central components of patent policy: namely, place inventions in the public domain once the patent has expired.[115] However, the Federal Circuit has severely curtailed both the enablement and best mode requirements for successful software patents.[116] Disclosure of source code is not required, nor is any disclosure regarding how to write the computer software.[117] Such limited disclosure prevents many of the invention's useful elements from reaching the public domain.

Second, any iterative attempt to innovate by improving already existing software likely infringes that software's patents. For example, the plain language of the Patent Act prohibits reverse engineering of software.[118] Every time a computer runs software or even decompiles software—a process critical for reverse engineering—it makes a copy of that software in the memory of the computer. This copy almost certainly constitutes infringement.[119] Patent holders can prevent any third-party or competitor from duplicating any part of software that contains a patented process because decompiling the software requires making a copy of the patented part and constitutes direct patent infringement.

This restriction on reverse engineering stifles innovation in software much more than other patentable inventions. The Patent Act does not prohibit inventors from reverse engineering other patented inventions in order to make improvements on them. In fact, inventors can file patents covering improvements to already patented inventions.[120] The structure of software and the way it runs on computers artificially restricts this type of iterative improvement. Furthermore, the typical software development cycle exacerbates this prohibition.

More than many other types of invention, software relies on rapid incremental improvements. Such iteration is not possible when developers must reinvent the wheel because disclosure as to the best mode of implementation of a software patent is lacking and they cannot reverse engineer to discover it. Additionally, under the doctrine of equivalents, a court may find infringement even if the accused process does not explicitly

[115] *See* Cohen & Lemley, *supra* note 100, at 24 n.86.

[116] Fonar Corp. v. Gen. Elec. Co., 107 F.3d 1543, 1549 (Fed. Cir. 1997) ("[W}here software constitutes part of a best mode of carrying out an invention, description of such a best mode is satisfied by a disclosure of the functions of the software.").

[117] *See* Cohen & Lemley, *supra* note 100, at 24 n.87.

[118] The defenses of first sale, implied license, experimental use, and patent misuse probably do not apply to defend a patent infringement claim alleging reverse engineering of software. *Id.* at 29–36.

[119] *Id.* at 19.

[120] 35 U.S.C. § 101 (2012) ("Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, *or any new and useful improvement thereof*, may obtain a patent therefor, subject to the conditions and requirements of this title.") (emphasis added).

fall within the patent but is substantially equivalent to the patented invention.[121]

These patentability issues have presented significant impediments to open systems. For example, Android relies on third parties to bring its software to market on their cell phone hardware. Any group that controls a process patent incorporated into Android can prevent a third party from running Android on its system. This is because with Android installation on a new phone, the third party is infringing on the software patent for a particular process, even if 99% of the Android software itself does not infringe any patents.[122] This scenario has already come to light in the form of patent licensing payments to Microsoft for each Android phone sold.[123] This problem is specific to open systems like Android because they rely on compatibility with other devices, which may require use of third-party processes to connect those devices.

Closed systems, on the other hand, do not rely as heavily on interoperability of third-party devices and are therefore much less likely to face this issue. iOS has not faced the same patent issues for several other reasons, the most significant being that it has been first in the market with many cell phone operating system innovations, and also because Apple has been particularly zealous in patenting its inventions.

Both open and closed systems face a myriad of antitrust and patent concerns. As discussed above, antitrust liability generally arises more frequently for closed systems while open systems face greater difficulties with patent infringement.

III. Proposed Regulations to Improve Innovation

The overview above outlines the current state of regulation of iOS and Android. Both Apple and Google have broad leeway to use antitrust laws and intellectual property rights to interfere with competitors and each other under the guise of exclusivity. Several changes to competition and intellectual property regulation could prevent these abuses.

The overriding theme of current regulation is that open systems tend to commit fewer anticompetitive harms but are more likely to

[121] Graver Tank & Mfg. Co. v. Linde Air Prods. Co., 339 U.S. 605, 608 (1950). *See* Warner-Jenkinson Co., Inc. v. Hilton Davis Chem. Co., 520 U.S. 17, 29 (1997).

[122] One may point out that this exact same scenario applies to every classic patentable process, like the process to create a pharmaceutical. There are two differences in the case of software patents from those processes. First, in this case, one patented process that comprises a miniscule proportion of the entire software is preventing companies from licensing an otherwise completely legitimate product. Second, the structure of the software injury, and particularly that of cell phone operating systems, does not warrant the strong patent protections available in other industries. By default, innovation in the software industry occurs at an iterative pace; use of a patent controlling a fairly limited process to control the dissemination of an entire product creates more harm than good.

[123] *See* Steve Lohr, *Microsoft's Patent Strategy Against Android*, Bits Blog N.Y. Times (Oct. 3, 2011, 12:20 PM), http://bits.blogs.nytimes.com/2011/10/03/microsofts-patent-strategy-against-android.

infringe on patented processes or violate the privacy of consumers. The balance of new regulation depends on whether one prioritizes innovation over property rights or vice-versa.

It is more important to promote generative systems than to uphold laws protecting the intellectual property regimes in place, especially in the software field. In other words, the law should promote open systems above patent protection in high tech fields. Patent law promotes innovation by incentivizing invention. Especially in the world of software, development often involves iterative changes and user collaboration. It is important to foster this type of growth in order to maintain the forward pace of innovation in software. Furthermore, the promotion of open systems prevents restriction of information and maintains technology as a vibrant space for expression of new ideas.[124] When systems are closed, society suffers because people are less able to develop their own modes of expression. Therefore, regulations of the industry in question should promote open, generative systems.

Several changes to both antitrust and patent law will promote generative systems. Chief among these is a change to the standard of proof in sham litigation cases in order to prevent patent holders from interfering with legitimate inventions similar to those covered by their patents.

A. Modernize the Sham Litigation Claim Through Use of Subjective Intent Evidence

As discussed above, the use of the litigation process in an anticompetitive manner can lead to severe penalties under the antitrust laws. The standard for succeeding in one of these claims greatly favors defendants. In order to successfully state an antitrust claim revolving around sham litigation, a plaintiff must show that the underlying litigation is both objectively and subjectively unreasonable.[125] Establishing objective unreasonableness requires showing that there is no probable cause to bring suit, a standard which means that any suit which has even a modicum of merit—even if strongly motivated by anticompetitive intent—will not lead to antitrust liability. There have been several recent proposals to expand the sham litigation exception to *Noerr-Pennington* immunity.

Congress has begun to recognize the need for reform of the objective/subjective test. At a recent hearing before the Subcommittee on Intellectual Property, Competition, and the Internet of the House of Representatives Committee on the Judiciary, several witnesses proposed liberalizing the objective/subjective test due to the difficulty of establishing

[124] *See* Jonathan Zittrain, *Protecting the Internet Without Wrecking It*, Boston Review, (Mar. 2008), http://bostonreview.net/forum/protecting-internet-without-wrecking-it.

[125] *See* Prof'l Real Estate Investors, Inc. v. Columbia Pictures Indus., Inc., 508 U.S. 49, 60 (1993).

lack of probable cause.[126] For example, evidence that executives of a company knew that a patent was invalid is part of the second subjective prong of the subjective/objective test. But under the current standard, a prospective plaintiff cannot use this information until it objectively proves that the litigation had no chance of success. This seems impossible if the Patent Office approves the patent, given that such approval is prima facie evidence of the patent's validity. The ban on subjective evidence that clearly shows culpability is irrational because it allows those who admit to filing sham lawsuits to avoid liability.

Liberalizing the sham litigation standard to permit the use of subjective evidence will significantly help promote open and generative systems. It will become more difficult for holders of questionable software patents[127] to use the bully pulpit to prevent others from using their discoveries, which are likely not in the public domain, to innovate. Under a subjective standard, if it is obvious that a company is using its intellectual property to foreclose competition without caring about protecting its property, it becomes much easier for smaller companies to defend themselves from sham suits because complete destruction of the intellectual property claims is not required to win the suit. Additionally it will significantly lower the effectiveness of patent trolls,[128] who rely on embellishing claims related to patents they received on assignment over inventions that they had no hand in creating.

Allowing subjective evidence at the threshold of litigation has its downsides. Any sort of subjective evidence places a greater burden on the courts to sift through evidence, creating a larger likelihood that litigation will cost more, take longer to complete, and lead to less conclusive results. These are valid concerns. However, lengthier litigation that is defensible on its merits is preferable to a system in which large firms can launch blatant sham litigation against smaller ones with no fear of losing because the litigation has a scintilla of probable cause.

Including subjective intent in this manner further promotes the goals of open and generative systems by deterring patent trolls and others who file sham litigation from proceeding with their claims in the first place. This frees smaller companies from the prospect of full-scale defenses of their products from tenuous-at-best infringement claims when such defenses may be prohibitively expensive.

[126] *See, e.g.*, *Litigation as a Predatory Practice: Hearing Before the Subcomm. on Intellectual Property, Competition, and the Internet of the Comm. on the Judiciary H.R.*, 112th Cong. 9–20 (2012) (statement of J. Douglas Richards, Partner, Cohen Milstein Sellers & Toll, PLLC).

[127] Software patents are much more likely to be questionable because they tend to represent smaller iterative improvements rather than breakthrough innovations, leading to obviousness concerns. *See* DAN L. BURK & MARK A. LEMLEY, THE PATENT CRISIS AND HOW THE COURTS CAN SOLVE IT 156–58 (2009).

[128] Patent trolls, also known as non-practicing entities, are patent-holders who acquire patents without the intent to use or manufacture the patented invention.

A more cogent criticism of the inclusion of subjective intent evidence at the outset of sham litigation lies in the fact that including subjective evidence at the outset of a case—before examining the objective basis of a suit—may prohibit firms from exercising their explicit rights under intellectual property laws to exclude others from using their inventions. Intellectual property, particularly a patent, grants the holder the complete right of exclusion to an invention for a period of time in order to recoup the invention's costs of research and development. The entire purpose of any infringement suit is to prevent another party from using a protected invention. This purpose in and of itself is anticompetitive, and any examination into the subjective intent of the complainant will show anticompetitive intent. All patent suits would then be sham litigation under the antitrust laws. This label would prevent intellectual property holders with monopoly power in certain industries from enforcing their valid rights to exclude others.

This criticism misses a key ingredient of any sham litigation under the antitrust laws. The counterclaiming party must still show that the infringement claims will fail and that the primary purpose of the claims was anticompetitive. Valid intellectual property claims are unaffected. In *Kodak II*, the Ninth Circuit outlined this framework as it applied to a refusal to deal.[129] The Ninth Circuit framework prevents a pretextual refusal to deal, even where valid patents support such a refusal.[130] Use of subjective intent would proceed similarly in a sham litigation claim. Valid intellectual property rights would still be fully enforceable, and even failed litigation would not be a sham without strong evidence of anticompetitive intent.[131] Promotion of generative systems requires reform of the objective/subjective test in order to prevent subjectively baseless patent infringement suits. As detailed below, promoting innovation requires several other patent reforms.

B. Reduce the Number of Patent Infringement Actions

The nature of the current software patent regime does not promote open and generative systems. Three fixes to the way courts interpret patent rights would promote significantly more innovation and

[129] Image Tech. Servs., Inc. v. Eastman Kodak Co., 125 F.3d 1195, 1219–20 (9th Cir. 1997).

[130] *Id.*

[131] The need to be sure that the intent behind alleged sham litigation is anticompetitive is strong enough to preserve First Amendment rights to petition the government for grievances in court. Any claim of sham litigation would need to be supported by clear and convincing evidence of anticompetitive intent, in addition to failed intellectual property claims in order to overcome the presumption in favor of free use of the judicial system without facing liability for doing so. *See* E. R.R. Presidents Conference v. Noerr Motor Freight, Inc., 365 U.S. 127, 138 (1961). *But see* Wolfe v. George, 486 F.3d 1120, 1125–26 (9th Cir. 2007) (upholding statute prohibiting vexatious litigants from filing suit because it was rationally related to a legitimate government interest, and false statements are not immunized by the First Amendment right to petition).

compatibility among software systems by reducing the number of patent infringement actions: (1) Allow reverse engineering of software patents; (2) permit an independent invention defense; and (3) import examination of intent evidence in regard to business improvements that close systems from access by competitors.

1. Allow Reverse Engineering of Software Patents

Allowing reverse engineering of software would dramatically increase innovation within the software industry for three reasons. First, allowing reverse engineering would free competitors to examine each other's improvements in detail without fear of patent infringement and improve upon them rather than imitate them. In fact, the iterative nature of innovation in the software industry requires reverse engineering. The inability to reverse engineer patented software forces developers to reinvent the wheel on every new project. Currently, the courts interpret the scope and latitude given to patents under the doctrine of equivalency by examining the degree of non-obviousness of the invention.[132] A more pioneering invention is entitled to a broader range of equivalence than others.[133] In other words, more significant inventions receive more protection under patent laws and a wider range of imitators are likely to infringe. But the standard pattern of innovation in the software industry involves considerable reuse of old code. Therefore, software is much less likely to be pioneering in nature and is likely to run into significant infringement issues under the doctrine of equivalency.

Allowing reverse engineering alleviates the problem of infringement under the doctrine of equivalency. Under the current regime, programmers must resolve old problems without reverse engineering and then, faced with time pressures to release frequent software updates, add small iterative improvements to their code. If inventors are able to reverse engineer software, they will be able to focus on making significant improvements without resolving old problems. Furthermore, allowing reverse engineering will help return the content and best mode of implementation of the patented software to the public domain, fulfilling the public policy of disclosure which drives patent grants.

Second, allowing reverse engineering would promote the creation of open, compatible, and generative systems because practitioners could to understand each formerly patented system in enough detail to create innovative compatible additions to formerly closed and walled-off software. For example, allowing reverse engineering of Apple's patented

[132] Cohen & Lemley, *supra* note 100, at 40.

[133] *Id.*

iOS code would allow competitors to make software that runs more efficiently and connects in new and unimagined ways with Apple devices.[134]

Finally, permitting reverse engineering of patents will dramatically reduce the number of patent infringement claims filed by competitor corporations, freeing up resources for other more efficient activities.

Allowing reverse engineering has its downsides. The primary danger is that competitor firms can discover how a patented product works and then use the protected knowledge to copy and create an infringing product. This argument draws little weight because in theory, there should be no need to reverse engineer any patented product. Theoretically, patenting an invention requires placing the know-how behind the invention in the public domain.[135] However, as discussed above, software patents require almost no disclosure of the best practices or methods behind the invention to the patent office. Concededly, it is unclear how to accomplish such a disclosure short of adding hundreds of technical pages to software patents. This makes an even stronger case that reverse engineering may be the only way to place patented software in the public domain. In fact, intellectual property law has long accepted reverse engineering as a means to advance science and technology. The underlying discoveries of intellectual property lie in the public domain and protections are in place to prevent profit from direct infringement of protected inventions. However, marketing an improved invention has never been actionable under patent laws because it runs counter to their purpose.[136] Allowing reverse engineering directly improves software as much as it improves innovation in every scientific field. Allowing reverse engineering for software puts it on an even playing field.

Established firms also worry that allowing reverse engineering of patented software will lead to higher levels of infringement because it will be easier to replicate complex algorithms in competitive software. Additionally, firms are concerned that the ability to reverse engineer with impunity will lead to significant security breaches, as those who look to invade secure software will have legal license to do so. These concerns ignore key aspects of patent and other legal protections. Patent protection is still valid after reverse engineering. The inventing party will still be able to sue any imitator for infringement derived from the reverse engineering, just not the act of reverse engineering itself. Second, removal of the protection against reverse engineering will not exacerbate any potential security issues because those who seek to breach software security systems

[134] Apple would no doubt argue that this type of access severely degrades the stability and security of the iOS environment. At the base level, there is a tradeoff amongst generativity, compatibility, and security. Open systems that foster innovation need to be maintained in order to drive the economy forward, and that security can be achieved with this in mind, not despite it.

[135] *See* 35 U.S.C. § 112 (2012).

[136] *See id.* § 101 (declaring improvements to be patentable subject matter).

are already breaching a whole host of other laws, including criminal statutes with more serious penalties than those imposed in patent infringement actions. Given these other laws, and the fact that they are significantly easier to prosecute than patent infringement cases, the prohibition on reverse engineering does not deter security breaches.

Finally, because the current limitation arises from the plain language of the patent laws, allowing reverse engineering in software patents would require a change to the statute, which might be difficult.

2. *Allow an Independent Invention Defense*

Allowing independent inventors limited rights to use their inventions will prevent abusive patent infringement actions. Many commentators have proposed an independent invention defense to patent infringement.[137] Under this proposal, any inventor who unknowingly creates an already patented invention prior to receiving actual or constructive notice[138] of the patent application will be immune from infringing on that patent. Congress enacted a more limited version of this proposal as part of the America Invents Act.[139] The prior use defense under the current regime requires that the claimant show prior commercial use of the invention at least one year before the date of the filing of the patent.[140] An independent invention defense would remove these one year and commercial use requirements.[141]

An independent invention defense makes economic sense. Studies have shown that a 10% price reduction to an invention charged by a patent monopoly will reduce the patentee's profits by 1% while decreasing the social costs of the patent monopoly by 19%.[142] This evidence suggests that existing patent monopolies are suboptimal in favor of protection to the patentee. Furthermore, the case for monopoly rights in the use and control of intellectual property similar to those that exist in the use and control of tangible property is weak because it is possible for two parties to use intellectual property equally efficiently due to its non-competitive

[137] *See, e.g.*, Samson Vermont, *Independent Invention as a Defense to Patent Infringement*, 105 MICH. L. REV. 475 (2006).

[138] Publishing the details of the invention in a public repository or industry publication constitutes notice if the re-inventor receives that publication before she re-invents the patented subject matter. Publication of the details of the invention in recognized scientific journals or presented at conferences also constitutes public notice. A blind posting of an invention where a re-inventor is not likely to come across it does not constitute public notice. *See id.* at 486.

[139] 35 U.S.C. § 273.

[140] *Id.*

[141] This defense is not especially relevant for the software industry due to the iterative process of software development and the short lead time between invention of a new patentable software process and its introduction to the market.

[142] *See* Ian Ayres & Paul Klemperer, *Limiting Patentees' Market Power Without Reducing Innovation Incentives: The Perverse Benefits of Uncertainty and Non-Injunctive Remedies*, 97 MICH. L. REV. 985, 989–90 (1999).

nature. Tangible property does not enjoy the same advantage.[143] Finally, the independent inventor will not be able to sue others for patent infringement or violation of other patent rights. Those rights remain squarely with the first inventor to patent.

Such a defense would eliminate the spurious patent infringement cases that have particular impact on the software industry.[144] In particular, patent trolls and other entities that do not manufacture or invent their own technology plague the software industry. An independent invention defense will prevent patent trolls from undertaking two specific practices that warrant concern: First, patent trolls will no longer be able to file continuations on patents that track the improvements in competitor's products because, by definition, competitors invented those products without prior knowledge of the patented product.[145] Patent trolls will be much less pernicious in the software industry where companies develop almost all improvements with short lead times, meaning that those improvements are all most likely developed independently and without knowledge of other patented material. Independent software developers would not be subject to suit by patent trolls who hold patents but do not influence development of improved software in any way.

Finally, an independent invention defense would allow software developers to independently create inventions without fear of patent infringement. Software development is not a difficult undertaking, but research into the patent pool to determine whether a software design is already prior art is nearly impossible for many of the small, independent development studios that drive software innovation.[146] Given this industry structure, it is likely that most software inventions are independent of one another. Because constructive notice requires publication in relevant media or presentation to the software community, research into the software patent pool will become simpler and more streamlined. All other independently created software will be free from infringement actions.

There are two primary obstacles to an independent invention defense. The first and most cogent is that first inventors will learn to issue notice immediately, foreclosing the independent invention defense. This is not necessarily objection to the defense; rather, it forces firms to publicize and disclose their inventions to the public in order to obtain maximum protection under the patent laws. This fulfills one of the primary policies behind patent protection: disclosure of inventions into the public domain. Critics are correct that increased notice will render the defense

[143] Vermont, *supra* note 137, at 477.

[144] *See* Mark A. Lemley, *Should Patent Infringement Require Proof of Copying?*, 105 MICH. L. REV. 1525, 1526 n.5 (2007).

[145] *See id.* at 1526.

[146] Christina Mulligan & Timothy B. Lee, *Scaling the Patent System*, at 13–14,(Mar. 6, 2012), http://ssrn.com/abstract=2016968.

moot, but the defense is not designed to allow all inventors who come in second place in an invention race to claim protection, it is designed to apply only to those who legitimately invent with no knowledge of the prior art.

Second, opponents argue that firms will fake reinvention in order to take advantage of the defense. This argument is unconvincing for two reasons: (1) Fraud in the patent world is incredibly rare; and (2) fraudulently reinventing a product has a much lesser reward than fraudulently patenting it in the first place because the re-inventor does not receive exclusionary rights.[147] Furthermore, re-inventors will have a stringent evidentiary burden to show that they reinvented before receiving notice. An independent inventor defense for software patents makes sense because of the structure of the industry and the high social costs of requiring programmers to reinvent the wheel every time they write a new program. This defense, combined with allowing reverse engineering, will considerably limit anticompetitive and meritless patent infringement actions.

C. Increase Scrutiny of the "Business Improvements" Rationale Through an Examination of Subjective Intent

Companies should not be able to mask anticompetitive changes to their products through a post hoc business improvement rationale. It is true that the law should give companies broad leeway to innovate without opening themselves up to antitrust liability under a predatory innovation scheme, but innovations masquerading as business improvements that eliminate compatibility with other devices should be subject to some antitrust scrutiny as opposed to none at all in order to promote more open and generative systems. Currently, under *Allied Orthopedic Alliances v. Tyco Health Care Group*, the courts find any test purporting to weigh the procompetitive and anticompetitive effects of product improvements both burdensome on innovation and impossible to administer.[148] However, this standard allows a company to get away with any business improvement. Those "improvements" clearly designed to destroy competitive interoperability under the guise of a marginal improvement in a product stoke the most concern. Such business improvements cause more anticompetitive harm than actual benefit.

The solution to preventing pretextual business improvements lies in an examination of intent. Just as with illegal tying involving patent protection, the courts should look to the intent of the monopolist when evaluating product improvements whose byproducts include significant

[147] *See* Vermont, *supra* note 137, at 502.

[148] 592 F.3d 991, 1000 (9th Cir. 2010).

reductions in compatibility with competitive products and services.[149] Of course, the burden of proof regarding anticompetitive intent lies with the complainant. A defendant will have the opportunity to assert a valid business justification and the plaintiff will be able to rebut it with positive evidence of the defendant's anticompetitive intent.

These situations are somewhat analogous to tying. Business improvements that reduce compatibility accomplish the same effects as tying without explicitly forbidding the non-tied products. Where tying requires a company to affirmatively link the tying and tied products together, a business improvement that reduces compatibility in effect forces linkage between the tying and tied product with the exact same result. The only difference is the purported justification, which in many cases could be licensed or modified to prevent the compatibility issues it creates while still providing the sought after improvement. In reality, predatory innovation involves purposeful modification of a product to reduce compatibility with competitive products.

In *In re Apple iPod iTunes Antitrust Litigation*, Apple modified iTunes to encrypt files in a different, more secure way and then subsequently modified iTunes again to prevent any other product from reading or writing to iPods.[150] The court found that the first change was a business improvement and refused to consider its anticompetitive effects, even though those effects were significant because they prevented compatibility between competitive products and the iPod.[151] The Court did not rule on the second change because there were genuine issues of fact regarding whether it was actually a business improvement.[152] However, the second change itself eliminated *all* compatibility between the iPod and competitive software. This severely anticompetitive action, used to foreclose competition in the market, goes unchecked under the current antitrust jurisprudence.

Intent is the key indicator in cases addressing business improvements. Courts have found tests regarding innovation non-administrable for good reason. It is nearly impossible to weigh the procompetitive effects of an innovation with the anticompetitive effects on competitors. Therefore, intent is a more effective gauge that allows courts to distill the anticompetitive actions of the parties. Courts are skilled at distilling and evaluating intent. Lawyers and judges have specific training to evaluate a person's credibility and to determine one's inner thoughts from her actions. In the most legislated branch of our legal system—criminal

[149] *See* Image Technical Servs., Inc. v. Eastman Kodak Co., 125 F.3d 1195, 1219 (9th Cir. 1997).

[150] 796 F. Supp. 2d 1137, 1143, 1146 (N.D. Cal. 2011).

[151] *Id.* at 1144.

[152] *Id.* at 1147. This was probably because Apple failed to preclude all genuine issues of material fact on summary judgment, not because they did not show what would eventually be a valid business justification.

law—almost every major crime requires proof of knowing intent. Our legal system has established methods by which to determine an actor's intent. There is no reason not to impute those skills into the antitrust realm.

Some commentators have decried the use of intent in antitrust on the grounds that intent proves too much.[153] In theory, competitors always intend to defeat each other, so an examination of the intent of competitors' actions will always show anticompetitive intent.[154] In practice, however, this is simply not the case. There is a huge gap between intentional anticompetitive actions to harm a competitor by eliminating compatibility and actions that harm a competitor through the introduction of a superior product. The latter is clearly a procompetitive action, the type that the antitrust laws seek to protect. The former does not rely on superior skill, industry, or foresight, but rather is a pure example of a company using monopoly power to foreclose competition by cutting off access to the market. An examination of intent can see through these anticompetitive actions in a way that is fair, equitable, and administrable by the courts.

Conclusion

The backbone of innovation in the software industry comes from the ability of users to grapple with, modify, and combine existing products to create iterative improvements. Android and iOS provide helpful examples of two differing tracks of software innovation. Both provide stable platforms for software developers to innovate in the mobile technology space. Continuing to promote innovation requires that small parties with few legal resources be permitted to continue innovation without facing anticompetitive threats from established players in the market and infringement actions from holders of questionable patents. This Note proposes amendments to the sham litigation doctrine under *Noerr-Pennington*, modifications to the patent regime allowing reverse engineering of software and an independent invention defense, and a modification to the business improvements justification under section 2 of the Sherman Act.

Each of these changes to the regulatory scheme for networked systems will increase openness and, by extension, innovation. They do so by (1) limiting the use of the courts to block invention in the name of patent rights held by trolls and other non-practicing entities, (2) reducing meritless and burdensome patent infringement litigation, and (3) disallowing pretextual conduct by established industry players to anticompetitively reduce compatibility with their products.

[153] *See* Michael A. Carrier, *Unraveling the Patent-Antitrust Paradox*, 150 U. Pa. L. Rev. 761, 794 (2002).

[154] *Id.*

***