

No. 13-298

IN THE
Supreme Court of the United States

ALICE CORPORATION PTY. LTD.,

Petitioner,

v.

CLS BANK INTERNATIONAL, ET AL.,

Respondents.

ON WRIT OF CERTIORARI TO THE UNITED STATES
COURT OF APPEALS FOR THE FEDERAL CIRCUIT

**BRIEF OF *AMICI CURIAE*
CHECKPOINT SOFTWARE, INC.,
COLLATERAL ANALYTICS, LLC, DAILY
MOTION, INC., GITHUB, INC., GUIDEWIRE
SOFTWARE, INC., HIPMUNK, INC., JIVE
SOFTWARE, INC., LINKEDIN, INC., MESSAGE
SYSTEMS, INC., NETFLIX, INC., NEWEGG,
INC., QUANTUM CORPORATION,
RACKSPACE, INC., RING CENTRAL, INC.,
TRULIA, INC., TWITTER, INC., AND YELP
INC. IN SUPPORT OF RESPONDENTS**

MARK A. LEMLEY
Counsel of Record
CLEMENT S. ROBERTS
DURIE TANGRI LLP
217 Leidesdorff Street
San Francisco, CA 94111
(415) 362-6666
mlemley@durietangri.com
Counsel for Amici Curiae

Stefani E. Shanberg
WILSON SONSINI
GOODRICH & ROSATI
Professional Corporation
One Market Plaza
Spear Street Tower,
Suite 3300
San Francisco, CA 94105
(415) 947-2000

*Counsel for Amici Curiae
Jive Software, Inc. and
Hipmunk, Inc.*

TABLE OF CONTENTS

	Page
INTEREST OF THE AMICI CURIAE	1
SUMMARY OF ARGUMENT	2
I. Software Patents Are Not Necessary to Drive Innovation	3
II. Software Patents Hinder Innovation	6
CONCLUSION.....	12

TABLE OF CITED AUTHORITIES

Page(s)

Statutes

17 U.S.C. § 101	5
-----------------------	---

Other Authorities

Adam Messinger, <i>Introducing the Innovator’s Patent Agreement</i> , Twitter Engineering Blog, (Apr. 17, 2012)	4
Carl Shapiro, <i>Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting</i> , 1 <i>Innovation Pol’y & Econ.</i> 119 (2000)	9
Catherine Tucker, <i>Patent Trolls and Technology Diffusion</i> (Tilburg Law & Econ. Ctr., Discussion Paper No. 2012-030).....	11
David A. Burton, <i>Software Developers Want Changes in Patent and Copyright Law</i> , 2 <i>Mich. Telecom. Tech. L. Rev.</i> 87 (1996)	3
David Drummond, <i>When Patents Attack Android</i> , Google: Official Blog (Aug. 3, 2011)	8

TABLE OF CITED AUTHORITIES

	Page(s)
Ed Sutherland, <i>WiMax, 802.11n Renew Patent Debate</i> , WiFi Planet (Apr. 7, 2005)	8
James Bessen & Michael J. Meurer, <i>Patent Failure: How Judges, Lawyers, and Bureaucrats Put Innovation at Risk</i> (2008)	3, 10
James E. Bessen et al., <i>The Private and Social Costs of Patent Trolls</i> (Boston Univ. Sch. of Law, Working Paper No. 11-45, 2011)	10
John R. Allison & Mark A. Lemley, <i>Empirical Evidence on the Validity of Litigated Patents</i> , 26 AIPLA Q.J. 185 (1998)	9
Josh Lerner et al., <i>The Dynamics of Open Source Contributors</i> , 96 Am. Econ. Rev. 114 (2006)	5

TABLE OF CITED AUTHORITIES

Page(s)

Karim R. Lakhani & Robert G. Wolf, Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects, in <i>Perspectives on Free and Open Source Software</i> 3 (Joseph Feller et al., eds. 2005).....	5
Mark A. Lemley & Carl Shapiro, <i>Patent Holdup and Royalty Stacking</i> , 85 Tex. L. Rev. 1991, 2028-29 (2007)	8
Mark A. Lemley, Address, <i>Software Patents and the Return of Functional Claiming</i> , 2013 Wis. L. Rev. 905	4, 7
Mark A. Lemley, <i>Ignoring Patents</i> , 2008 Mich. St. L. Rev. 19.....	9
Michael A. Heller & Rebecca S. Eisenberg, <i>Can Patents Deter Innovation? The Anticommons in Biomedical Research</i> , 280 Sci. 698 (1998).....	9

TABLE OF CITED AUTHORITIES

Page(s)

Robert E. Purvy, *Software
Obviousness: The Disconnect
Between Engineers and the
Patent System* (2014)..... 3

Wendy Seltzer, *Software Patents
and/or Software Development*,
78 Brooklyn L. Rev. 929 (2013)..... 7, 9

Rules

Supreme Court Rule 37.6 1

INTEREST OF THE AMICI CURIAE¹

Amici are small and medium-sized companies that develop and sell software and a variety of products that use and incorporate software. Amici are among the most innovative and fastest-growing companies in the world, and collectively own many software patents. Amici have no direct interest in the outcome of this case, but have a strong interest in seeing that the patent system encourages, and does not impede, innovation.

¹ Counsel for the parties have filed blanket consents to the filing of amicus briefs. Pursuant to Rule 37.6, no counsel for either party had any role in authoring this brief in whole or in part, and no party other than the named Amici has made any monetary contribution toward the preparation and submission of this brief.

SUMMARY OF ARGUMENT

Software patents do not serve the Constitutional purpose of the patent system: to promote the progress of science and the useful arts. Software companies, including the undersigned, do not innovate in hopes of obtaining software patents. Rather, we create innovative software because of our desire to delight our customers and despite, not because of, the patent system.

Amici take no position in this brief on what the law of patentable subject matter requires, or on whether software should be patentable. But some have suggested to this Court that software patents are necessary to support innovation. They are not. The undersigned believe that innovation happens despite software patents, not because of them.

ARGUMENT

I. Software Patents Are Not Necessary to Drive Innovation

Software patents are not necessary to spur innovation among the Amici. Our engineers do not innovate because they hope to get patents. To the contrary, our programmers (like programmers generally) overwhelmingly oppose software patents.² “Surveys regularly find that computer programmers are opposed to patents on software by a wide margin.” James Bessen & Michael J. Meurer, *Patent Failure: How Judges, Lawyers, and Bureaucrats Put Innovation at Risk* (2008). One early study found that “[b]y lopsided margins ranging from over 2-to-1 to more than 10-to-1, the software developers said that software patents . . . impede software development and should be abolished.” David A. Burton, *Software Developers Want Changes in Patent and Copyright Law*, 2 Mich. Telecom. Tech. L. Rev. 87 (1996), available at http://www.mttl.org/voltwo/burton_art.html.

² See, e.g., Robert E. Purvy, *Software Obviousness: The Disconnect Between Engineers and the Patent System* at 1 (2014), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2399580 (“Software engineers overwhelmingly dislike software patents and consider nearly all of them obvious.”).

Indeed, companies such as Twitter have begun to compete for engineering talent by offering to restrict offensive patenting, making their patents difficult for patent trolls to use or enforce in court. *See, e.g.*, Adam Messinger, *Introducing the Innovator's Patent Agreement*, Twitter Engineering Blog, (Apr. 17, 2012), at <https://blog.twitter.com/2012/introducing-innovators-patent-agreement>. The fact that companies are competing for inventors by promising to reduce this type of patenting suggests that software patents do not motivate those inventors.

Nor do software patents serve the goal of financing innovation. Our investors do not invest with the hopes of acquiring patents, but with the hope of creating successful products. Product life cycles move very quickly in software. Software patents, by contrast, take over three years on average to issue, *see* Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 Wis. L. Rev. 905, 930, and are often enforced years after that issuance. Thus, by the time a software patent issues – and certainly by the time it is enforced – it is likely to be directed at obsolete technology. This obsolescence causes many of the problems frequently seen in software patent litigation, including the fact that software patent litigation often involves efforts to read vague, functional claim language onto features and functions that did not exist at the time the patent application was filed. This dynamic also means that patents are singularly ineffective at protecting software products from competitors and therefore do

not serve as a substantial driver of innovation in the software industry.

Instead, a variety of non-patent incentives spur innovation in the software industry. First-mover advantages can provide substantial rewards to innovators by giving them a jump on the market, leaving imitators scrambling to catch up. Network effects may allow innovators to capture significant returns because the value of the software often increases with the number of users, and thus users want to use the program that everyone else is using. The entire open-source ecosystem exists despite patents because its programmers strive for personal satisfaction, reputation, and a desire to solve a problem. See, e.g., Karim R. Lakhani & Robert G. Wolf, *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects*, in *Perspectives on Free and Open Source Software* 3, 11-12 (Joseph Feller et al., eds. 2005). Amici and other companies benefit by employing those programmers to build and improve their products. See Josh Lerner et al., *The Dynamics of Open Source Contributors*, 96 *Am. Econ. Rev.* 114, 115 (2006). And both trade-secret and copyright law already protect software and effectively prevent both wrongful use and explicit copying by others. 17 U.S.C. § 101. Thus, inventors and investors who care about maximizing their profits do not necessarily need the incentives theoretically provided by patents.

Software patents, then, serve at most as a sort of consolation prize, giving unsuccessful innovators a

chance at recovering some portion of their investment years later. But investors invest in new companies and technologies because they hope to be successful – not because they hope to use software patents to extract royalties from more successful enterprises in the event of failure. To put it differently, inventors and investors are motivated by the hope of success – not by the prospect that they might (through litigation) recoup some portion of their investment in the event of failure. And software patents do not contribute to achieving success for most software companies.

II. Software Patents Hinder Innovation

Patent law tends to hinder rather than spur innovation in the software industry. The many flaws with software patents create a well-known patent thicket that discourages inventors and companies from investing in new research, development, and products.

Software patents are problematic because of the uncertainty associated with the meaning and scope of their claims, which are often drafted in functional language and without a shared vocabulary. Accordingly, it is often impossible to know what a software patent covers until after a court has construed the patent's claims and the parties have spent months on litigation. Even worse, patentees can often benefit from ambiguous patent claims by twisting the language of the patent claim to cover

something the inventor did not have in mind at the time.³ Indeed, because computer technology changes so quickly, and it takes more than three years to get a patent out of the PTO on average, software patents are almost always asserted against technology that is several product generations removed from the patentee's invention, compounding the problem of trying to understand the scope of software patents.

Software patents are often phrased to claim high-level, functional ideas. But an idea is only a small part of a successful software program. Wendy Seltzer, *Software Patents and/or Software Development*, 78 Brooklyn L. Rev. 929, 949 (2013) (“Software developers treat as commonplace the proposition that the idea is only a small part of the final product. . . . It is the implementation that remains difficult and time-consuming.”). Successful software companies implement that idea, developing the functioning software in an iterative process that results in constant improvement to the code. As Wendy Seltzer puts it, “[i]nstead of inducing optimal development of innovative software, the rules of the patent system encourage the production of patents.” *Id.* at 943.

³ Although much of the broad, functional, and ambiguous claim language should be eliminated by a patent examiner during prosecution of the patent, that is often not the case. See Lemley, *Software Patents and the Return of Functional Claiming*, *supra* (documenting the widespread use of broad functional claims in software patents).

These problems are compounded by the large number of software patents. Estimates vary widely, in part because it is hard to define a software patent, but there are certainly hundreds of thousands of them, many covering trivial variations of routine functionality. But because computer products – as opposed to patents – inevitably integrate complex, multicomponent technology, any given product is potentially subject to a large number of patents. For example, 3G wireless technology was subject to more than 7,000 claimed “essential” patents as of 2004; that number is doubtless much higher now. See Mark A. Lemley & Carl Shapiro, *Patent Holdup and Royalty Stacking*, 85 Tex. L. Rev. 1991, 2028-29 (2007). WiFi is subject to hundreds and probably thousands of claimed essential patents. See Ed Sutherland, *WiMax, 802.11n Renew Patent Debate*, WiFi Planet (Apr. 7, 2005), <http://www.wifiplanet.com/columns/article.php/3495951>. And the problem is even worse than these numbers suggest, since both 3G wireless technology and WiFi are not themselves products but merely components that must be integrated into a final product. Some industry experts have estimated that 250,000 patents go into a modern smartphone. David Drummond, *When Patents Attack Android*, Google: Official Blog (Aug. 3, 2011), <http://googleblog.blogspot.com/2011/08/when-patents-attack-android.html> (statement of David Drummond, Chief Legal Officer at Google). The result is what Carl Shapiro has called a “patent thicket” — an overwhelming set of overlapping patent rights that impede innovation. See Carl Shapiro, *Navigating the Patent Thicket: Cross*

Licenses, Patent Pools, and Standard Setting, 1 Innovation Pol’y & Econ. 119, 120 (2000). See also Michael A. Heller & Rebecca S. Eisenberg, *Can Patents Deter Innovation? The Anticommons in Biomedical Research*, 280 Sci. 698 (1998).

Further, many asserted software patents are invalid. Empirical evidence suggests that nearly half of all asserted patents are invalid, John R. Allison & Mark A. Lemley, *Empirical Evidence on the Validity of Litigated Patents*, 26 AIPLA Q.J. 185, 205 (1998) (forty-six percent of litigated patent claims invalid), and software patents appear (to the Amici) to be more likely than most to be invalid. Thus, even if an innovative, product-producing software company acting in good faith could identify the thousands of patents that might be held to read on its product, any effort to license those patents would have to involve paying for thousands of invalid patents.

It is no wonder that “[s]oftware developers do not perform patent searches before developing products.” Seltzer, *supra*, at 955. Indeed, research suggests that companies in the software industry largely ignore patents unless and until they are threatened with suit – and that this behavior is economically rational. See Mark A. Lemley, *Ignoring Patents*, 2008 Mich. St. L. Rev. 19, 21-22. The fact that we have created a system in which it makes no sense to read software patents undermines the disclosure benefit long thought to be at the heart of the patent system.

It also encourages litigation. Indeed, it is virtually certain that if a software product is successful, its maker can expect to be hit with dozens of suits and receive hundreds of threat letters from patent owners.⁴ These lawsuits are expensive, draining resources that would otherwise be put towards innovation. They are also socially costly. According to one estimate, patent trolls cost the economy \$500 billion over the last twenty years, mostly in the information technology industry. James E. Bessen et al., *The Private and Social Costs of Patent Trolls* at 17 (Boston Univ. Sch. of Law, Working Paper No. 11-45, 2011), http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1930272. Elsewhere, the same authors find that patents in the information technology industry have a net negative effect on the market value of companies in the industry. Bessen & Meurer, *Patent Failure*, *supra*, at 137. Other reports suggest that patent trolls inhibit innovation at the firms they sue.

⁴ To take just a few examples, Lex Machina data shows that as of May 1, 2012, Apple had been named in 298 patent lawsuits over the last dozen years, Microsoft in 269 patent lawsuits, Google in 151, Yahoo! in 91, Oracle in 58, Facebook in 56, SAP in 38, Yelp in 9, and Twitter in 8. Lex Machina, <http://www.lexmachina.com>. While some of these companies, notably Apple and Oracle, are plaintiffs in some suits, the overwhelming majority of these cases involve the named companies as patent infringement defendants, and the majority are filed by patent trolls.

Catherine Tucker, *Patent Trolls and Technology Diffusion* at 4 (Tilburg Law & Econ. Ctr., Discussion Paper No. 2012-030), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1976593. That is a far cry from what the patent system is supposed to do: encourage innovation.

CONCLUSION

We do not express an opinion on what the law of patentable subject matter requires. Nor do we suggest that patents could never be appropriate for software. But the software patent system as it is currently designed does not promote innovation, but rather hinders it.

Respectfully submitted,

MARK A. LEMLEY
Counsel of Record
CLEMENT S. ROBERTS
DURIE TANGRI LLP
217 Leidesdorff Street
San Francisco, CA 94111
(415) 362-6666
mlemley@durietangri.com
Counsel for Amici Curiae

Stefani E. Shanberg
WILSON SONSINI
GOODRICH & ROSATI
Professional Corporation
One Market Plaza
Spear Street Tower,
Suite 3300
San Francisco, CA 94105
(415) 947-2000
*Counsel for Amici Curiae
Jive Software, Inc. and
Hipmunk, Inc.*

February 27, 2014