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*
LAW, BUSINESS, AND ECONOMICS SCHOLARS
IN SUPPORT OF RESPONDENTS**

————————◆————————

JASON M. SCHULTZ
(*Counsel of Record*)
NYU TECHNOLOGY LAW
 AND POLICY CLINIC
NYU SCHOOL OF LAW
245 Sullivan Street, 609
New York, NY 10012
Telephone: (212) 992-7365
jason.schultz@
 exchange.law.nyu.edu

BRIAN J. LOVE
Assistant Professor of Law
SANTA CLARA UNIVERSITY
 SCHOOL OF LAW
500 El Camino Real
Santa Clara, CA 95053
Telephone: (408) 554-4783
blove@scu.edu

Counsel for Amici Curiae
Law, Business, and Economics Scholars

i

TABLE OF CONTENTS

Page

TABLE OF CONTENTS – Continued

iii

TABLE OF AUTHORITIES

Page

TABLE OF AUTHORITIES – Continued

TABLE OF AUTHORITIES – Continued

Page

TABLE OF AUTHORITIES – Continued

TABLE OF AUTHORITIES – Continued

TABLE OF AUTHORITIES – Continued

INTEREST OF *AMICI CURIAE*

Amici are professors and scholars who teach and write on legal and economic issues and are concerned about the role of patent law in promoting technological innovation. They are Timothy K. Armstrong, James E. Bessen, Michele Boldrin, Irene Calboli, Brian W. Carver, Ralph D. Clifford, Wesley M. Cohen, Eric Goldman, Brad A. Greenberg, Bronwyn H. Hall, Christian Helmers, Karim R. Lakhani, David K. Levine, Brian J. Love, Eric S. Maskin, Michael J. Meurer, Shawn P. Miller, Connie Davis Nichols, Tyler T. Ochoa, Jorge R. Roig, Matthew Sag, F. M. Scherer, Jason M. Schultz, Katherine J. Strandberg, Alexander Tabarrok, and Eric von Hippel. Various *amici* have taught, researched, and published analyses on the role of patent law as an incentive to inventors and entrepreneurs. A summary of the qualifications and affiliations of the individual *amici* is provided at the end of this brief, though it should be noted that *amici* file this brief solely as individuals and not on behalf of the institutions with which they are affiliated. *Amici* represent neither party in this action and offer the following views on this matter.[1]

————————◆————————

---

[1] In accordance with S. Ct. R. 37.3(a), all parties have consented to the filing of this brief. The Petitioner and Respondents have filed consent letters with the Clerk. Pursuant to S. Ct. R. 37.6, counsel for *amici* state that no counsel for a party authored this brief in whole or in part, and no person or entity other than *amici* or their counsel made a monetary contribution to the preparation or submission of this brief.

SUMMARY OF ARGUMENT

The Federal Circuit's expansion of patentable subject matter in the 1990s led to a threefold increase in software patents, many of which contain abstract ideas merely tethered to a general-purpose computer. There is little evidence, however, to suggest this expansion has produced an increase in software innovation. The software industry was highly innovative in the decade immediately prior to this expansion, when the viability of software patentability was unclear and software patents were few. When surveyed, most software developers oppose software patenting, and, in practice, software innovators tend to rely on other tools to capture market share such as first-mover advantage, trade secrecy, copyright, goodwill, and economic network effects. If anything, the increase in software patenting has led to an increase in software litigation, which in turn has encouraged firms to acquire patents for strategic purposes unrelated to innovation, serving as either defensive stockpiles to deter legal threats or offensive leverage for rent-seeking patent assertion entities (PAEs).

Moreover, abstract software patents do not function well within a property rights framework because they fail to define cognizable metes and bounds and fail to provide effective notice to third parties of when a particular practice or product might infringe. Due to their abstractness, these claims can often be construed to cover any of the particularized processes

that result in the same outcome, *including those never envisioned by the inventor*. Accordingly, these metes and bounds are not concrete enough to be useful to those who wish to tread carefully around them. The mere application of the idea using general-purpose technological components, such as a general-purpose computer, does nothing to abate this problem. Similarly, abstract patents defy the attempts of software innovators, or general counsel at technology companies, to stay on notice of what is already protected. This leaves firms vulnerable to investing in software development with little to no assurance that they will be able to avoid infringing upon an abstract patent, even if they conduct diligent searches within patent databases. Again, this will be true even if there are general-purpose technological components tethered to the claims, as those components do nothing to help distinguish one abstract claim from another. Proliferation of such patents also contributes to the problem of patent thickets.

A well-defined 35 U.S.C. § 101 ensures that abstract software patent claims and their attendant notice and patent thicket problems do not undermine the patent system and stymie innovation. It serves as a decisive gatekeeper that the Patent Office and trial courts can use early in administrative proceedings and litigation. Further, it avoids many of the systemic challenges prevalent with the use of 35 U.S.C. §§ 102, 103, and 112 in such cases – the speed of software innovation, the difficulty

locating software prior art, and lax, broad claiming standards. Accordingly, this Court should affirm the invalidity of the patent claims at issue here and hold that abstract ideas in the form of software are unpatentable and that mere computer implementation of those ideas does not create patentability.

————◆————

ARGUMENT

The Constitution empowers Congress to create a patent system granting exclusive rights to inventors, but only as a means of encouraging innovation. *See* U.S. Const. art. I, § 8, cl. 8. This Court's jurisprudence has consistently recognized that extending patent protection to abstract ideas not only fails to increase innovation overall, but threatens to impede its progress. *See Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1293 (2012) ("[M]onopolization of [abstract ideas] through the grant of a patent might tend to impede innovation more than it would tend to promote it."); *Bilski v. Kappos*, 130 S. Ct. 3218, 3229 (2010). Nonetheless, the U.S. Court of Appeals for the Federal Circuit has gradually expanded the scope of patentable subject matter for abstract ideas over the last quarter century, culminating with its holdings in *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994) and *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998). As a result of this misguided expansion, the patent system has become increasingly unbalanced and even hostile in some respects to

high technology entrepreneurs and inventors. This case presents the Court with an opportunity to expand on its efforts in *Bilski* and *Mayo* to restore balance by reaffirming the abstract idea exclusion as a robust gatekeeper that prohibits abstract patent claims, such as those asserted in this case, no matter the form in which their drafters attempt to claim them. *Amici* present this brief and the empirical evidence within it in support of such an opinion.

I. Abstract Software Patents Discourage Innovation.

In *Gottschalk v. Benson*, 409 U.S. 63 (1972) as well as *Parker v. Flook*, 437 U.S. 584 (1978), this Court recognized the risks of allowing patents on the abstract formulas and concepts contained in software even when they were tied tangentially to a physical apparatus. *Flook*, 437 U.S. at 590 (extending this principle by holding that post-solution activity does not "transform an unpatentable principle into a patentable process," otherwise any "competent draftsman could attach some form of post-solution activity to almost any mathematical formula."); *Benson*, 409 U.S. at 68 (invalidating claims for being "so abstract and sweeping as to cover both known and unknown uses" of a method of numerical conversion on a general-purpose computer). When this Court did find such a patent valid in *Diamond v. Diehr*, 450 U.S. 175 (1981), its holding was limited to the application of a formula in a specific industrial process. *Id*. at 192. Unfortunately the Federal Circuit

misunderstood this delicate analysis and increasingly expanded the scope of patentable subject matter, culminating with its decisions in *Alappat* and *State Street*, allowing otherwise abstract patent claims to qualify under § 101 based on a mechanical "magic words" approach. *State Street*, 149 F.3d at 1375 (requiring that an abstract software claim merely produce a "useful, concrete and tangible result"); *Alappat*, 33 F.3d at 1545 (holding that a software program based on an abstract idea was patentable as long as the claim included a general-purpose computer).

This expansion has contributed to a dramatic increase in the number of patents related to software, both in absolute terms and as a percentage of issued patents. *See* U.S. Gov't Accountability Office, GAO-13-465, *Intellectual Property: Assessing Factors That Affect Patent Infringement Litigation Could Help Improve Patent Quality* 13 (2013), *available at* http://1.usa.gov/1gatCRr (noting that after *Alappat* and *State Street*, there was a substantial increase in the number of patents granted with software claims). The following chart demonstrates this increase:

Number of patents
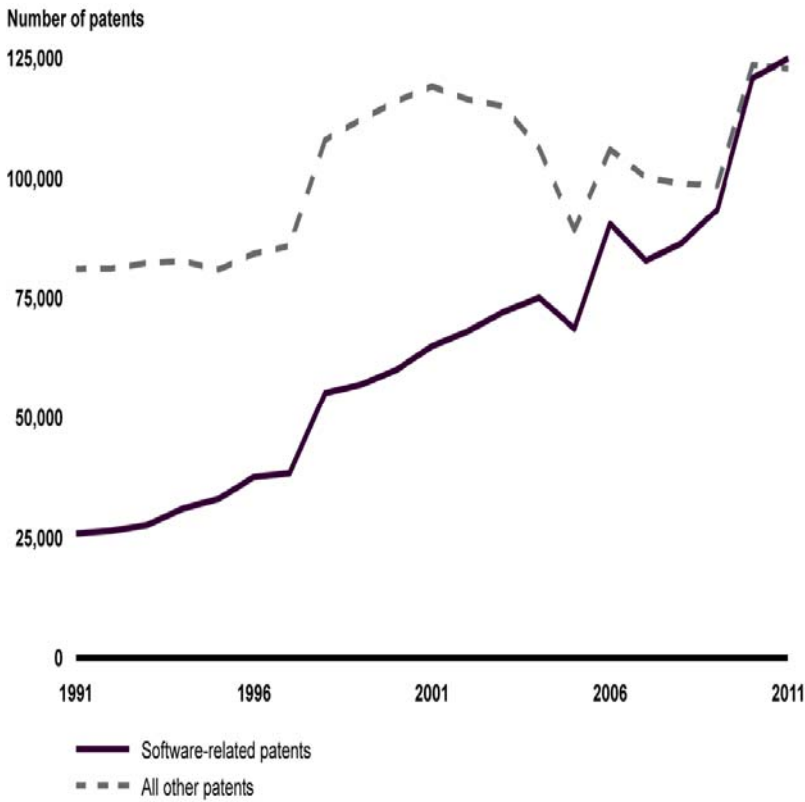
Id. at 12. Take, for example, the estimated 11,000 patents covering the sale of goods online. James Bessen & Michael J. Meurer, *Patent Failure: How Judges, Bureaucrats, and Lawyers Put Innovators at Risk* 213 (2008). These patents emerged, at least in part, because otherwise unpatentable abstract ideas – e.g., holding desired products in a shopping cart, checking out upon shopping completion, pressing a button to "buy it now" – became the patentable inventions of the first patent applicant to suggest

implementing the idea on a computer. *Cf. Alappat*, 33 F.3d at 1545.

A. Abstract Software Patents Should Be Limited, as There Is Little Evidence that Software Innovators Rely on Patenting for Incentives to Innovate and Compete.

Despite the expanded availability of patent protection, studies of the software industry have failed to show any corresponding increase in innovation as a result. While the number of software patents has increased, there is evidence that software entrepreneurs rely much less on patent incentives when building and maintaining competitive businesses than other factors, such as first mover advantage and trade secret, copyright, or trademark protection. *See* Stuart J.H. Graham et al., *High Technology Entrepreneurs and the Patent System: Results of the 2008 Berkeley Patent Survey*, 24 Berkeley Tech. L.J. 1255, 1288-90 (2009). In fact, studies suggest that spending on stronger and broader software patents is *negatively* correlated with spending on R&D. *See* Robert Hunt & James Bessen, *The Software Patent Experiment*, 77(3) Fed. Res. Bank Phila. Bus. Rev. 22, 27-29 (2004) ("[T]he negative correlation between increases in firms' focus on software patents and their R&D intensity in the 1990s suggests that firms may be substituting for R&D with software patents."); *see also* James Bessen

& Robert M. Hunt, *An Empirical Look at Software Patents* 30-33 (Fed. Reserve Bank of Phila. Working Paper No. 03-17, 2004) (reporting the results of a historical study showing that in the 1990s firms increased their propensity to patent after subject matter restrictions on software patents were relaxed, and that this increase in spending on patents displaced spending on R&D to such an extent that, assuming perfect substitution, R&D would have been about 10%, or $16 billion, higher if there had been no increase in software patenting).

History also suggests that patent incentives have had a negligible impact on software development overall. Despite the denial of software patentability in *Benson* and *Flook*, which rejected broad-based abstract claims, or after *Diehr*, which only allowed patentability involving software in a very narrow concrete context, software innovation flourished in the period between *Benson* and *Alappat*. *See* Martin Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry* 18-19 tbl.1.2 (2003) (noting that the software industry grew eightfold between 1980 and 1990, with revenues increasing from $6.1 billion to $51.3 billion). In addition, surveys of software developers during this period show most were, in fact, *opposed* to patents. *See, e.g.*, Effy Oz, *Acceptable Protection of Software Intellectual Property: A Survey of Software Developers and Lawyers,* 34 Info. & Mgmt. 161, 167-70 (1998); Pamela Samuelson et al., *Developments on the Intellectual Property Front*, 35(6) Comms. of the ACM 33,

35 (1992); Bessen & Meurer, *Patent Failure*, *supra*, at 189 ("[S]uch broad opposition from *within* the affected industry and among the affected inventors seems to be unprecedented in U.S. patent history."). And, despite the fact that software patents have been growing both in absolute terms and as a percentage of all patents granted in recent decades, most software firms still do not patent their software products. James Bessen, *A Generation of Software Patents*, 18 B.U. J. Sci. & Tech. L. 241, 255 (2012). This appears to be even more true for startup firms. *Id*. at 255-56.

Indeed, practically speaking, it is unlikely that patent rights can spur innovation in software because:

> [I]nnovation in the high-tech industry is exceedingly fast, while government bureaucracy is exceedingly slow. Computing power doubles roughly every two years, but it generally takes three to four years to receive a patent, and even longer to enforce it in court. As a result, patented inventions in the high-tech sector are invariably yesterday's news.

Brian J. Love, *No: Software Patents Don't Spur Innovation, but Impede It*, Wall St. J., May 12, 2013, at R2.

B. The Need for Socially Wasteful Defensive Bulwarks Has Primarily Driven Software Patenting Since *Alappat* and *State Street*, Rather than the Desire to Protect Investments in R&D or to Promote Competition.

While there is little evidence that expanded software patent eligibility has spurred innovation, it is increasingly apparent that it has spurred patent litigation. *See* Bessen, *supra*, at 261 ("[T]he number of lawsuits involving software patents has more than tripled since 1999."); Bessen & Meurer, *Patent Failure*, *supra*, at 193 ("Software patents issued in more recent years are much more likely to be litigated, not less."). This increase is not merely in proportion to the growing number of software patents; the probability that any single software patent will result in litigation has also been rising over time, meaning that both defendants and software patentees face a higher risk of litigation and business uncertainty. The following chart demonstrates this:
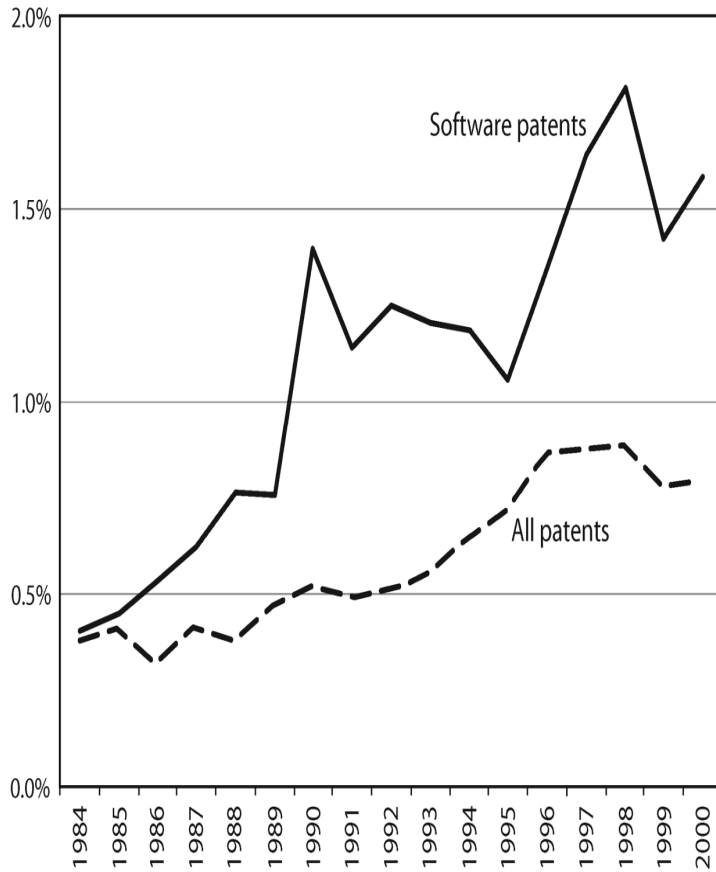
Probability that a Patent Is in Lawsuit Within Four Years of Issue

Note: Data adjusted for under-reporting.

Bessen & Meurer, *Patent Failure*, *supra*, at 193, fig.9.2. The increase in software patent litigation is such that the GAO reports that the increase in overall patent litigation is largely attributable to the

growth of litigation over software patents. U.S. Gov't Accountability Office, *supra*, at 14.

Ironically the increase in litigation has led many operating software companies to obtain patents to fend off suits, rather than to file them. *See* Colleen V. Chien, *From Arms Race to Marketplace: The Complex Patent Ecosystem and Its Implications for the Patent System*, 62 Hastings L.J. 297, 306-10 (2010) (describing the practice of defensive patenting by high-tech firms as growing rapidly after the late 1990s); Jonathan Masur, *Costly Screens and Patent Examination*, 2 J. Legal Analysis 687, 704-14 (2010) (explaining how patents with high private value – defined as the value of a patent as a tool for initiating or deterring litigation – often fail to generate socially beneficial research); Gideon Parchomovsky & R. Polk Wagner, *Patent Portfolios*, 154 U. Pa. L. Rev. 1, 26 (2005) (discussing how patents can be more valuable to companies in aggregate as a defensive portfolio than individually as reward mechanisms). In this context, patents no longer function to reward innovation: they are merely an added cost of doing business, an insurance policy against litigation as opposed to a reward for productive R&D. For larger firms this practice of amassing stockpiles of broad patents has amounted to a Cold War-era arms race that produces costly deterrence through an implied threat of mutually assured destruction by litigation. *See generally* Chien, *From Arms Race to Marketplace: The Complex Patent Ecosystem and Its Implications for the Patent System*, *supra*, at 333-34; Colleen V. Chien,

Race to the Bottom, 51 Intell. Asset Mgmt. Mag. 10 (2012). The costs of this practice are incurred on multiple levels: first, the cost to the firm and its shareholders of acquiring the defensive patent; second, the cost of defending against a software patent lawsuit; and third, the cost to society that occurs when innovation is chilled or firms are deterred from entering a market. Some large technology companies have devoted billions of dollars to building their patent portfolios, spending more on strategic patent acquisitions and litigation than on overall R&D. *See* Charles Duhigg & Steve Lohr, *The Patent, Used as a Sword*, N.Y. Times (Feb. 25, 8:14 PM), http://nyti.ms/N0Ba0T; *see also* Steven Church, Tim Culpan & Devin Banerjee, *Apple Joins Microsoft, RIM in $4.5 Billion Buy of Nortel Patents*, Bloomberg (July 1, 2011, 12:18 PM), http://bloom.bg/1aLJegC; James Kendrick, *Google Buys Motorola Mobility and Its Patent Portfolio for $12.5 Billion*, ZDNet (Aug. 15, 2011 5:27 PM), http://zd.net/1iE21KY.

C. The Explosion of Patent Assertion Entity Activity Has Aggravated the Costs of Abstract Software Claims.

A significant contributor to the rise in software patent litigation is the rise of patent assertion entities (PAEs).[2] PAEs are firms that obtain patents

---

[2] Patent assertion entities are sometimes called non-practicing entities (NPEs) or patent trolls, though in the cited studies both refer to the same type of firm.

without intent to support or develop new products; instead, their business model is based on threatening costly litigation against practicing firms to extract settlements and licensing fees. Exec. Office of the President, *Patent Assertion and U.S. Innovation* 1 (2013), http://1.usa.gov/1oExPD1. The fact that PAEs do not actually use the patents they own outside of litigation effectively immunizes them from counter-claiming and the defensive patent strategies discussed above. *Id*. at 4. In 2013 the White House released a report by the President's Council of Economic Advisers, the National Economic Council, and the Office of Science & Technology Policy that concluded on the basis of an extensive review of empirical studies that PAEs have had an overall negative impact on innovation and economic growth. *Id* at 2.[3] A recent study further suggests that the burden imposed by PAEs is disproportionately borne by small and medium sized firms who are more likely to be threatened with litigation by PAEs than large firms.

---

[3] As a general matter PAEs follow three different strategic approaches when extracting rents: 1) a "lottery ticket" model where PAEs acquire broad, abstract patents in the hopes of obtaining outsized jury awards, 2) a "bottom-feeder" model where the PAEs actually avoid litigation, relying instead on the general high cost of patent litigation to obtain many small settlement agreements, or 3) building massive portfolios of patents to extort licensing fees from practicing firms. *See* Mark A. Lemley & A. Douglas Melamed, *Missing the Forest for the Trolls*, 113 Colum. L. Rev. 2117, 2126-28 (2013). The latter two strategies are much less likely to result in litigation than the lottery ticket model, so PAEs in such cases have little incentive to care about the quality of the patents they obtain. *Id*.

James Bessen & Michael J. Meurer, *The Direct Costs from NPE Disputes,* 99 Cornell L. Rev. 387, 398-99 (2014) (reporting survey results that firms with less than $1 billion in annual revenues made up 90% of the sampled defendants in PAE lawsuits between 2005-2011, and firms with less than $100 million in revenues made up 82% of all defendants, despite the fact that these small and medium sized firms only made up 44% of the overall survey sample).

PAEs are particularly relevant to the software industry because the bulk of PAE litigation involves software patent claims, which are particularly susceptible to abstraction problems. *See* James Bessen, Jennifer Ford & Michael J. Meurer, *The Private and Social Costs of Patent Trolls*, 34(4) Regulation Mag. 26, 29 (2011-2012) (reporting empirical study findings that 62% of patents litigated by PAEs between 1990 and 2010 were "software patents" and 75% covered "computer and communications technology"); *see also* Brian J. Love, *An Empirical Study of Patent Litigation Timing: Could a Patent Term Reduction Decimate Trolls Without Harming Innovators?*, 161 U. Pa. L. Rev. 1309, 1347 (2013) (finding that roughly 80% of PAE lawsuits were brought to enforce claims with high-tech subject matter, 65% of which were software-related claims); U.S. Gov't Accountability Office, *supra*, at 22 ("Specifically, about 84 percent of [PAE] lawsuits from 2007 to 2011 involved software-related patents, while about 35 percent of operating company lawsuits did."). PAE activity has increased so rapidly that in 2012 over half of all patent litigation cases could be attributed to PAEs. Exec. Office of the

President, *supra*, at 5. It is estimated that companies spent approximately $29 billion in 2011 dealing with these lawsuits. *See* Bessen & Meurer, *The Direct Costs from NPE Disputes, supra*, at 389. Apple, Samsung, HP, AT&T, Dell, Google, and Amazon.com were each sued by PAEs thirty or more times in 2013 alone. *Most Pursued Companies*, PatentFreedom, http://bit.ly/M5egUQ (last visited Feb. 23, 2014).

D. A Robust Application of § 101's Abstract Idea Prohibition Restores the Balance that *Alappat* and *State Street* Disrupted.

This Court has stated repeatedly that the prohibition on the patenting of abstract ideas is central to maintaining a proper balance in the patent system. *Mayo*, 132 S. Ct. at 1301 ("The Court has repeatedly emphasized . . . a concern that patent law not inhibit further discovery by improperly tying up the future use of laws of nature."); *Benson*, 409 U.S. at 67 ("Phenomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work."). The Court has an opportunity to return patent law to the course it established in *Benson*, *Flook*, and *Diehr*, and further reaffirmed in *Bilski* and *Mayo*, by ensuring that § 101 provides a robust gatekeeping function to prohibit abstract software claims, including those whose only tether to physical reality is the computer itself or other similar general-purpose technological devices. *Bilski*, 130 S. Ct. at 3230.

Opponents to this Court's position in *Bilski* and *Mayo* often claim that such holdings harm companies who have spent substantial amounts of money relying on the Federal Circuit's holdings in *Alappat* and *State Street*. Such claims, however, are misguided. Given the evidence above, the large amount of money invested in software patents has not led to more innovation and sales growth; instead, they have been a part of the costly practice of building defensive patent portfolios to stave off litigation and have even come back to penalize software firms through greater PAE activity. A robust § 101 gatekeeper would mitigate these costs and free software firms' capital from defensive patent stockpiling, making it available for productive and innovative R&D. This would further diminish the problem of patent thickets (discussed *infra*, §II(C)), ultimately increasing competition in the software industry and lowering cost of entry for innovative new startup companies. *See* Iain M. Cockburn & Megan J. MacGarvie, *Entry and Patenting in the Software Industry,* 57 Mgmt. Sci. 915, 926-29 (2011).

II. Section 101 Ensures the Patent System Operates as an Efficient Property Rights System.

In addition to skewing incentives and increasing socially wasteful spending, abstract software patents have undermined the patent system's ability to function as an efficient property rights system because they (1) fail to define clear metes and bounds

for claims, (2) fail to provide effective public notice to avoid inadvertent infringement, and (3) contribute to troublesome patent thickets.

A. Abstract Software Patents Fail to Define Clear Metes and Bounds.

Unlike the physical landscape, the "patentscape" lacks unique and recognizable geological features to serve as common points of reference when describing the shape and size of an entitlement. The metes and bounds of a patent must, therefore, be interpreted by mapping the words in a claim to a range of technologies that are sufficiently analogous to the invention actually possessed by the inventor. An abstract claim, however, fails to provide these metes and bounds, not because it is too broad, but because it "claims technologies unknown to the inventor." Bessen & Meurer, *Patent Failure*, *supra*, at 199. When an activity is described in an abstract manner – e.g. "the art of cutting ice by means of any power other than human power," *Wyeth v. Stone*, 30 F. Cas. 723, 727 (C.C.D. Mass. 1840) – the limits of the claim cease to be cognizable as some range of extant technologies. Thus, although the ice-cutter's patent was granted in the 1840s, there is no sensible mode of interpreting the boundaries of the claim that wouldn't extend to ice-cutting by laser, robot, or genetically modified polar bear. As Justice Story argued, "No man can have a right to cut ice by all means or methods, or by all or any sort of apparatus, although he is not the

inventor of any or all of such means, methods, or apparatus." *Id.*

Biological or chemical inventions will, at the very least, have a unique formula or compound molecule to serve as a point of reference. Software, on the other hand, may be capable only of functional, result-oriented definitions – e.g., cutting ice – which are amenable to broad interpretations that transcend the actual technologies originally known to the inventor. *See generally* Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 Wis. L. Rev. 905. Three examples, each litigated after *Alappat*, illustrate this interpretive ease. U.S. Patent No. 4,528,643, commonly known as "the Freeny patent," was granted for a vending machine that produced digital music tapes in stores, but the claim was cunningly drafted to be extremely comprehensive: a "system for reproducing information in material objects at a point of sale location." Freeny filed his application well before the advent of personal computers, e-commerce, and portable media players. Nonetheless, the company that came to hold the patent, E-DATA, alleged that purchasing, downloading, and transferring music from home computers onto media players or CD-ROMs were all within the patent's scope. *See Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323 (Fed. Cir. 2001). Similarly Wang Labs developed and patented an early operating system whose graphical user interface displayed information in "frames." Wang sued Netscape and America Online because these services

also displayed information inside of "frames," a feature those programs share with almost all modern user interfaces. *Wang Labs., Inc. v. Am. Online, Inc.*, 197 F.3d 1377 (Fed. Cir. 1999). Finally, in 2003, a PAE, Pinpoint, sued Amazon alleging that by recommending books to users, Amazon had infringed a patent on "systems recommending TV programs to viewers based on past choices." *Pinpoint, Inc. v. Amazon.com, Inc.*, 369 F. Supp. 2d 995 (N.D. Ill. 2005).

Such abstract claims as "displaying data in frames," "recommending media based on past choices," "reproducing information in material objects at a point of sale," or, as in the present case, using "a third party . . . to eliminate 'counterparty' or 'settlement' risk," simply cannot be reliably construed to define a reasonable area of covered technology. *See Wang*, 197 F.3d at 1379; *Interactive Gift*, 256 F.3d at 1323; *Pinpoint*, 369 F. Supp. 2d at 995; *cf. CLS Bank Int'l v. Alice Corp. Pty. Ltd.*, 717 F.3d 1269, 1274 (Fed. Cir. 2013). A general counsel at a technology startup would be hard-pressed to describe any concrete bounds or permissible follow-on innovations to her fellow engineers in the face of such claims. Any software that resulted in a similar functional result could be construed as infringing, and any investment in the commercialization of those technologies could inevitably carry liabilities, risks, and costs whose magnitudes are impossible to predict in advance. Thus, the property system that ostensibly exists to assure investors that long-term rents are secure does

the very opposite, casting a pall of uncertainty over the viability of any commercial product that happens to be adjacent to a lurking abstract claim.

B. Abstract Software Patents Fail to Provide Effective Public Notice.

Like metes and bounds, notice is also a concept the patent system borrows from traditional property law. Bessen & Meurer, *Patent Failure*, *supra*, at 206-07. The concepts are linked in so much as uncertainty surrounding borders often makes it impossible for citizens to be *on notice* and effectively plan their behavior with regards to navigating those borders or developing the immediately adjacent properties. Under patent law a citizen may fail to have notice of a claim not merely because the borders of the claim were difficult to discern, but because the claim as a whole was never imagined to be the stuff of patents. Such is the plight of a technology firm's general counsel who belatedly discovers that there are abstract claims that actually apply to longstanding industry conventions, "the basic tools of scientific and technological work." *Gottschalk v. Benson*, 409 U.S. 63, 67 (1972). This was surely the rude awakening suffered by counsel at Netscape and America Online when their companies were accused of infringing a decade-old patent on "displaying information in frames," despite the ubiquitous and unlicensed use of "frames" – windows or boxes – in software design. *Wang*, 197 F.3d at 1381. When sizable investments in product design, manufacture, and marketing are

made well before notice of a relevant abstract patent arrives, the inadvertently infringing manufacturer is in an extremely weak negotiating position. The patent holder can seek exorbitant royalties under credible threat of shutting down the manufacturer and rendering their prior efforts into fruitless sunk costs. *See* Carl Shapiro, *Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting, in* 1 *Innovation Policy and the Economy* 119, 125 (Adam B. Jaffe et al. eds., 2001).

Of course, one might argue that the job of a general counsel is to stay *on notice,* and so we may feel little pity for her failures. Perhaps she should have spent more time searching patent databases or monitoring the Patent & Trademark Office's (PTO) weekly publications for relevant claims. However these searches have costs, and if these search costs are too onerous, productive activity will quickly cease. The costs of staying on notice in the software patent system are very large because of the staggering number of patents to discover and read. *See* Bessen & Meurer, *Patent Failure*, *supra*, at 213 ("David M. Martin has estimated that 'if you're selling online, at the most recent count there are 4,319 patents you could be violating. If you also planned to advertise, receive payments for or plan shipments of your goods, you would need to be concerned with approximately 11,000.'"). Further, the cost in work hours of running the search can be extremely high. *See id*. ("One software executive estimates that checking clearance costs about $5,000 per patent."); Christina Mulligan

& Timothy B. Lee, *Scaling the Patent System*, 68 N.Y.U. Ann. Surv. Am. L. 289, 311-12 (2012) ("The more abstract an invention is, the more different parties are likely to use it for different purposes, and the more flexibility parties will have to describe it. All of these factors mean that more abstract patents will produce particularly high discovery costs."). If 11,000 patents need to be searched to engage in e-commerce while remaining on notice, and each patent will require a $5,000 investment in time and resources, then the careful software startup will be spending $55 million in capital before they can even begin to operate. Good news for lawyers, perhaps, though certainly not for in-house counsel, engineers, and others who depend on notice to ensure freedom to operate and innovate in new technologies.

Strategic PAE behavior also significantly increases the costs of staying on notice. Normally, even if one could not identify patent risks from searching for relevant claim language, one might search the patent holdings of known competitors to steer clear of potential infringement. However, PAEs are by definition non-competing entities and often assign their patents to shell companies and subsidiaries making an essential component of notice – *who owns what –* difficult to ascertain. For example the PAE Intellectual Ventures distributes its 25,000-50,000-deep patent portfolio among as many as 1,100 shell companies. Avancept, *A Study of: The Intellectual Ventures Portfolio in the United States: Patents & Applications* 6 (2d ed. 2010); *see also* Tom Ewing &

Robin Feldman, *The Giants Among Us*, 2012 Stan. Tech. L. Rev. 1, ¶¶ 21, 23, http://stanford.io/1jMBvPn. Similar behavior is widely reported across the PAE industry. *See generally* Colleen V. Chien, *The Who Owns What Problem in Patent Law* 3 (Santa Clara Univ. Legal Studies Research Paper No. 03-12, 2012). Obscured ownership allows PAEs to maintain an element of surprise, revealing the patent only after substantial investments have been made in the commercialization of an infringing technology. *Id.* While abstract patent claims are not the sole source of these strategic behaviors, they match obscure ownership with ambiguous claims yielding more effective ammunition for a patent ambush.

The strategic filing of continuation claims on pending abstract patents also undermines the patent system's notice function. Through continuation filings, patents can unfold gradually into an extended "family" of interrelated claims because patent applicants perpetually retain the right to abandon an application and restart the process, as well as the right to seek additional broader claims arising from the same application. Mark A. Lemley & Kimberly A. Moore, *Ending Abuse of Patent Continuations*, 84 B.U. L. Rev. 63, 64 (2004). This allows patents to linger in the PTO, lying in wait for new companies and technologies to emerge as potential defendants. For example, patents accompanied by continuation filings average 4.16 years before issuance as compared with solitary applications, which average 1.96 years. *Id*. at 71. This behavior is quite prevalent: 23%

of all patent applications contain continuances and 52% of all litigated patents had applications with continuances. *Id*. at 70. While the overall problem of continuations is beyond the scope of the question presented in this case, it is relevant to consider, given that the most litigated patents tend to be software patents owned by PAEs, *see* John R. Allison, Mark A. Lemley & Joshua Walker, *Extreme Value or Trolls on Top? The Characteristics of the Most-Litigated Patents*, 158 U. Pa. L. Rev. 1, 29-31 (2009), and that these highly litigated patents take greatest advantage of continuations, *see* Michael Risch, *Patent Troll Myths*, 42 Seton Hall L. Rev. 457, 479 (2012) (finding an average of four continuations per patent in a sample of the most litigated patents). Thus, a robust § 101 gatekeeper would provide a considerable check on such practices and the notice problems they create with respect to abstract software continuation claims.

C. Abstract Software Patents Contribute to Troublesome Patent Thickets.

When qualitative patent problems – boundary and notice – couple with the quantitative flood of abstract software patents, the result is a patent system so extensively partitioned and overlayed with exclusive rights that it stifles the very progress it is designed to promote. As a complicating factor, software innovation regularly relies on assembling smaller complementary innovations into cumulative products. The Internet browsers challenged by the

Wang patent exhibited data in "frames" as one of many components in a larger and complex way of displaying web pages. When a single product can potentially infringe thousands of patents, the expense of negotiating a license for each claim can become cost-prohibitive. Being an essential part of a whole, any single rights holder may choose to hold-out for the full value of the cumulative invention. *See* Mark A. Lemley & Carl Shapiro, *Patent Holdup and Royalty Stacking*, 85 Tex. L. Rev. 1991, 1994-98 (2007). Add to this the problem of abstraction, and the task of identifying all relevant rights and negotiating all appropriate licenses for each new version of every software product one sells becomes nearly impossible.

Scholars refer to the problem of overlapping rights as a patent thicket. *See generally* Bronwyn H. Hall et al., *A Study of Patent Thickets*, Report for the UK Intellectual Property Office (Oct. 2012), *available at* http://bit.ly/N6hAQt; Shapiro, *supra*. Patent thickets occur in other complex technologies and a number of public and private policies can ameliorate their effects. However, patents on abstract ideas exacerbate the effects of patent thickets. Even the conscientious inventor determined to push through a software thicket is likely to miss at least some abstract patents, especially if all patentees must do to qualify under § 101 is tether the abstract idea to a general-purpose computer or some other ubiquitous physical device. *See* Christopher A. Cotropia & Mark A. Lemley, *Copying in Patent Law*, 87 N.C. L. Rev. 1421, 1424 (2009) (finding that less than 3% of patent

litigation in the computer and software industries allege copying – i.e. 97% of cases involve allegations of innocent infringement). Unavoidable inadvertent infringement burdens innovators with unnecessary transaction costs, subverting the rule of law and the intent of the patent system. As Thomas Hobbes urged, "the use of laws is . . . as hedges are set, not to stop travellers, but to keep them in the way. . . . Unnecessary laws are not good laws, but traps for money." Thomas Hobbes, *Leviathan* 252 (A. R. Waller ed. 1904).[4]

III. Sections 102, 103, and 112 Are Not Replacements for the Role of § 101.

Some commentators, including the Federal Circuit, have made a case for allowing "substantive

---

[4] It is worth noting that more than efficiency is at stake when an overabundance of law fails to generate notice. Erratic enforcement and recondite doctrine prevent ordinary citizens from reliably planning their lives over the long term. The rule of law demands that citizens be treated as autonomous agents, capable of understanding rules and modifying their behavior voluntarily. Expediently coercing human behavior without first allowing for self-correction violates underlying principles of human dignity. Rational agents, citizens, are treated as cattle to be herded or horses to be broken. *See generally* Jeremy Waldron, *How Law Protects Dignity*, 71 Cambridge L.J. 200 (2012). A patent law that allows abstract claims destroys notice of essential legal commands; it treats society's brightest as valuable victims for the traps set by overzealous patentees and sophisticated, unscrupulous lawyers.

conditions of patentability" – § 102 (novelty), § 103 (nonobviousness), and § 112 (written description, enablement, and best mode) – to resolve the issue of systematically overbroad abstract software patents. *See, e.g.*, *MySpace, Inc. v. GraphOn Corp.*, 672 F.3d 1250, 1260 (Fed. Cir. 2012). However, properly construed, § 101 serves a unique and distinct role from §§ 102, 103, and 112 and, in many cases, a more appropriate one for addressing the incentive, boundary, notice, and thicket problems discussed above.

A. Sections 102, 103, and 112 Are Uniquely Ill-Suited to Correct Notice Problems in Software Patents.

In the context of software patents, §§ 102, 103, and 112 suffer from severe systemic challenges, including being ill-suited to reduce notice uncertainty and resulting in excessive litigation costs. *See* Brian J. Love, *Why Patentable Subject Matter Matters for Software*, 81 Geo. Wash. L. Rev. Arguendo 1, 6-10 (2012). For instance, in order to properly evaluate a claim under §§ 102 and 103, it must already have sufficiently clear metes and bounds by which to compare the scope of the claim against the prior art. Thus to function properly, §§ 102 and 103 rely on a robust § 101 to filter out abstract claims as a threshold matter.

Even when the scope of the claims is manageable the PTO has historically found it challenging to search for prior art within the software arts under

§§ 102 and 103. *See* Mark A. Lemley & Bhaven Sampat, *Examiner Characteristics and Patent Office Outcomes*, 94 Rev. Econ. & Stat. 817, 817-18 (2012). For one thing, PTO searches for prior art are generally limited to U.S. and foreign registered patent databases and commercial databases. Sean Tu, *Luck/Unluck of the Draw: An Empirical Study of Examiner Allowance Rates*, 2012 Stan. Tech. L. Rev. 10, ¶ 14, http://stanford.io/1h1YRjw. There is good reason to believe that much software prior art is not formally published anywhere, let alone in a patent application, meaning software prior art may never reveal itself in a search. *See* Robert P. Merges, *As Many as Six Impossible Patents Before Breakfast: Property Rights for Business Concepts and Patent System Reform*, 14 Berkeley Tech. L.J. 577, 589 (1999); *see also* Iain M. Cockburn & Megan J. MacGarvie, *Patents, Thickets and the Financing of Early-Stage Firms: Evidence from the Software Industry*, 18 J. Econ. & Mgmt. Strategy 729, 731-32 (2009). Lack of publication is also due to the amount of software innovation done outside of established research communities by parties who do not ordinarily make use of the patent system. *See* Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Calif. L. Rev. 1, 42 (2001). Innovations in unconventional fields are not necessarily described in published journals, but rather, exist in actual business methods or the source code of products that are available to consumers. *See id.* at 13; *see also* Bessen & Meurer, *Patent Failure*, *supra*, at 212-13 ("The general purpose nature of software

technology – again, because the technology is abstract, similar techniques can be used in a wide range of applications – means that techniques known in one realm might be applied in another, yet the documentary evidence that the Federal Circuit requires for a demonstration of obviousness might not be published.").

Even when software prior art *has* been published, it is difficult to search for and locate. The PTO itself lacks the resources or expertise to keep up with new prior art. *See* Note, *Estopping the Madness at the PTO: Improving Patent Administration Through Prosecution History Estoppel*, 116 Harv. L. Rev. 2164, 2171 (2003). Where software has been patented, the PTO's classification system has not historically been well-equipped to handle it. Cohen & Lemley, *supra*, at 13. Software innovations are difficult to describe and there is no standardized language for describing them that is known to all interested parties. As a result, performing a search for the words "software" or "computer" could turn up thousands of existing patents but still be missing thousands more that are relevant. Evidence, thus, shows that the PTO routinely issues software patents that overlook prior art. Merges, *supra*, at 589.

Section 112 also suffers from inadequacies as applied to software because the disclosure requirements for software inventions have proven exceptionally lax and permit extremely broad claiming. This makes it almost impossible to invalidate a software patent claim on disclosure grounds, as general

functional descriptions are considered adequate. Bessen & Meurer, *Patent Failure*, *supra*, at 210. Software "inventors" can, therefore, obtain claims so broad that they effectively patent the problem, rather than the solution. *See* Lemley, *supra*, at 928-36 (discussing how software patents manage to skirt the limits imposed on functional claiming).

B. Sections 102, 103, and 112 Generate Greater Litigation Costs than § 101.

The substantive conditions of patentability in §§ 102, 103, and 112 also have much greater litigation costs than § 101. While §§ 102 and 103 *require* the court to undertake claim construction as a predicate to conducting an invalidity analysis, § 101 generally does not require time-intensive claim construction to determine abstractness. *See, e.g., Ultramercial, Inc. v. Hulu, LLC*, 722 F.3d 1335, 1339 (Fed. Cir. 2013) ("[C]laim construction may not always be necessary for a § 101 analysis."); *Bancorp Servs., L.L.C. v. Sun Life Assur. Co. of Can. (U.S.)*, 687 F.3d 1266, 1273 (Fed. Cir. 2012) ("[W]e perceive no flaw in the notion that claim construction is not an inviolable prerequisite to a validity determination under § 101."). In fact, this Court has never required claim construction as a predicate for any of its § 101 opinions. *See, e.g., Bilski v. Kappos*, 130 S. Ct. 3218, 3222 (2010) (holding the patent to be ineligible on subject-matter grounds without claim construction). Moreover, an analysis under § 101 is less likely to involve factual

determinations, which would result in less discovery and a shorter timeframe for summary judgment.

District courts would also be well served by a strong, clear § 101. District courts have wide latitude in the order in which they decide issues, and providing clear guidance on deciding cases under § 101 would allow lower courts to address this issue and decide cases more quickly. *See CLS Bank Int'l v. Alice Corp. Pty. Ltd.*, 717 F.3d 1269, 1284 (Fed. Cir. 2013) ("District courts are rightly entrusted with great discretion to control their dockets and the conduct of proceedings before them, including the order of issues presented during litigation."). As it stands, the perception that § 101 lacks clarity hinders the broad use of this mechanism in litigation. *See MySpace*, 672 F.3d at 1260 (holding that courts may require litigants to address a patent's validity under §§ 102, 103, and 112 before reaching § 101, because doing so would avoid "the murky morass that is § 101 jurisprudence"); Love, *Why Patentable Subject Matter Matters for Software*, *supra*, at 4.

————◆————

CONCLUSION

Section 101 serves an essential role in our patent system. It works to ensure the proper administration of property rights by balancing incentives for innovation and providing adequate notice while maintaining public access to the ideas necessary for competition. This Court has long been familiar with the corrosive

effects of poorly designed intellectual property systems that "embaras[s] the honest pursuit of business with fears and apprehensions of concealed liens and unknown liabilities to lawsuits and vexatious accountings for profits made in good faith." *Atl. Works v. Brady*, 107 U.S. 192, 200 (1883). This case presents an opportunity to reaffirm the innovator's faith in a patent system that grants appropriate reward without abstract encumbrance. Accordingly, this Court should affirm the invalidity of the patent claims at issue here and further hold that abstract ideas in the form of software are unpatentable and that mere computer implementation of those ideas does not create patentability.

Respectfully submitted,

JASON M. SCHULTZ
(*Counsel of Record*)
NYU TECHNOLOGY LAW
  AND POLICY CLINIC
NYU SCHOOL OF LAW
245 Sullivan Street, 609
New York, NY 10012
Telephone: (212) 992-7365
jason.schultz@
  exchange.law.nyu.edu

BRIAN J. LOVE
Assistant Professor of Law
SANTA CLARA UNIVERSITY
  SCHOOL OF LAW
500 El Camino Real
Santa Clara, CA 95053
Telephone: (408) 554-4783
blove@scu.edu

Counsel for Amici Curiae
Law, Business, and Economics Scholars

APPENDIX

Amici Curiae Signatories*

Timothy K. Armstrong
Associate Dean of Faculty and Professor of Law, University of Cincinnati College of Law

James E. Bessen
Lecturer in Law, Boston University School of Law

Michele Boldrin
Joseph Gibson Hoyt Distinguished Professor in Arts & Sciences, Washington University in St. Louis

Irene Calboli
Professor of Law, Marquette University Law School

Brian W. Carver
Assistant Professor, University of California, Berkeley, School of Information

Ralph D. Clifford
Professor of Law, University of Massachusetts School of Law

Wesley M. Cohen
Frederick C. Joerg Professor of Business and Professor of Strategy, Economics, and Law, The Fuqua School of Business, Duke University

---

* Institutions are listed solely for identification purposes.

Eric Goldman
Professor of Law, Santa Clara University School of Law
Director, High Tech Law Institute, Santa Clara University

Brad A. Greenberg
Intellectual Property Fellow, Kernochan Center, Columbia Law School

Bronwyn H. Hall
Professor of the Graduate School, University of California, Berkeley

Professor of Economics of Technology and Innovation, University of Maastricht

Research Associate, National Bureau of Economic Research

Christian Helmers
Assistant Professor of Economics, Santa Clara University Leavey School of Business

Karim R. Lakhani
Lumry Family Associate Professor of Business Administration, Harvard Business School

David K. Levine
John H. Biggs Distinguished Professor, Washington University in St. Louis

Brian J. Love
Assistant Professor of Law, Santa Clara University School of Law

Eric S. Maskin
Adams University Professor, Harvard University
Nobel Memorial Prize in Economic Sciences, 2007

Michael J. Meurer
Abraham and Lillian Benton Scholar and Professor of Law, Boston University School of Law

Shawn P. Miller
Olin-Searle Fellow in Law, Science and Technology, Stanford Law School

Connie Davis Nichols
Associate Professor of Law, Baylor University School of Law

Tyler T. Ochoa
Professor of Law, Santa Clara University School of Law

Jorge R. Roig
Assistant Professor of Law, Charleston School of Law

Matthew Sag
Professor of Law, Loyola University Chicago School of Law

Associate Director for Intellectual Property of the Institute for Consumer Antitrust Studies

F. M. Scherer
Aetna Professor Emeritus, John F. Kennedy School of Government, Harvard University

Jason M. Schultz
Associate Professor of Clinical Law, New York University School of Law

Katherine J. Strandberg
Alfred B. Engelberg Professor of Law, New York University School of Law

Alexander Tabarrok
Bartley J. Madden Chair in Economics, George Mason University

Professor of Economics, George Mason University

Eric von Hippel
Professor of Technical Innovation, MIT Sloan School of Management

_____