

No. 13-298

In the Supreme Court of the United States

ALICE CORPORATION PTY., LTD.,

Petitioner,

v.

CLS BANK INTERNATIONAL & CLS SERVICES LTD.,

Respondent.

**On Writ of Certiorari to the United States
Court of Appeals for the Federal Circuit**

**BRIEF OF BSA | THE SOFTWARE ALLIANCE
AS *AMICUS CURIAE* IN
SUPPORT OF RESPONDENT**

ANDREW J. PINCUS

Counsel of Record

PAUL W. HUGHES

Mayer Brown LLP

1999 K Street, NW

Washington, DC 20006

(202) 263-3000

apincus@mayerbrown.com

Counsel for Amicus Curiae

TABLE OF CONTENTS

Table of Authorities.....	iii
Interest of the <i>Amicus Curiae</i>	1
Summary of Argument.....	1
Argument.....	4
I. Society Has Reaped Enormous Benefits From Computer Software Patents.....	4
A. Software Contributes Substantially To The U.S. Economy.	4
B. The Continued Patentability Of Software Is Critical To Economic Growth.	8
II. Software Innovations Are Eligible For Patent Protection.....	10
A. Software Is Patent-Eligible As Both A “Process” And As A “Machine.”	11
B. Software Innovations Are Not Categorically Exempt By The Implied Exceptions To Section 101.	14
C. <i>Stare Decisis</i> Precludes A Categorical Bar On Software Patentability.....	19
III. Section 101 Serves An Important, But Limited, Screening Function.....	20
A. Section 101 Need Not And Should Not Incorporate Other Patent Act Requirements.	21
B. Section 101’s Standards Apply To Software Claims In The Same Manner As Claimed Inventions In Other Fields Of Endeavor.....	24

TABLE OF CONTENTS—continued

1. A Software Algorithm Unrelated To A Law Of Nature, Physical Phenomenon, Or Abstract Idea Is Necessarily Eligible For Patent Protection Under Section 101.....	25
2. A Software Claim That Does Implement A Law Of Nature, Physical Phenomenon, Or Abstract Idea Is Patent Eligible If It Does “Significantly More” Than Describe The Patent-Ineligible Concept.	27
IV. The Claims In This Case Are Not Patent Eligible.	30
A. Intermediated Settlement Is An Abstract Idea.	30
B. The Patents Contain No Other Meaningful Limitations.	33
C. The Form Of The Claims Does Not Alter The Result.	35
Conclusion	36

TABLE OF AUTHORITIES

CASES

<i>Accenture Global Servs., GmbH v. Guidewire Software, Inc.</i> , 728 F.3d 1336 (Fed. Cir. 2013)	29
<i>In re Alappat</i> , 33 F.3d 1526 (Fed. Cir. 1994)	11, 13, 14
<i>Aristocrat Techs. Australia Pty Ltd. v. Int’l Game Tech.</i> , 521 F.3d 1328 (Fed. Cir. 2008)	24
<i>Ass’n for Molecular Pathology v. Myriad Genetics, Inc.</i> , 133 S. Ct. 2107 (2013)	24
<i>Bancorp Servs., L.L.C. v. Sun Life Assur. Co.</i> , 687 F.3d 1266 (Fed. Cir. 2012)	29
<i>In re Bernhart</i> , 417 F.2d 1395 (C.C.P.A. 1969)	11
<i>Bilski v. Kappos</i> , 130 S. Ct. 3218 (2010)	<i>passim</i>
<i>Blackboard, Inc. v. Desire2Learn, Inc.</i> , 574 F.3d 1371 (Fed. Cir. 2009)	23
<i>Bonito Boats, Inc. v. Thunder Craft Boats, Inc.</i> , 489 U.S. 141 (1989)	22
<i>CyberSource Corp. v. Retail Decisions, Inc.</i> , 654 F.3d 1366 (Fed. Cir. 2011)	29
<i>Dann v. Johnston</i> , 425 U.S. 219 (1976)	21

TABLE OF AUTHORITIES—continued

<i>Dealertrack, Inc. v. Huber</i> , 674 F.3d 1315 (Fed. Cir. 2012)	29
<i>Diamond v. Chakrabarty</i> , 447 U.S. 303 (1980)	13, 14
<i>Diamond v. Diehr</i> , 450 U.S. 175 (1981)	<i>passim</i>
<i>Eolas Techs. Inc. v. Microsoft Corp.</i> , 399 F.3d 1325 (Fed. Cir. 2005)	11
<i>Festo Corp. v. Kinzoku Kogyo Kabushiki Co.</i> , 535 U.S. 722 (2002)	20
<i>Finisar Corp. v. DirecTV Grp., Inc.</i> , 523 F.3d 1323 (Fed. Cir. 2008)	24
<i>Fort Props., Inc. v. Am. Master Lease LLC</i> , 671 F.3d 1317 (Fed. Cir. 2012)	29
<i>In re Freeman</i> , 573 F.2d 1237 (C.C.P.A. 1978)	11
<i>Gottschalk v. Benson</i> , 409 U.S. 63 (1972)	14, 17, 18, 19
<i>Halliburton Oil Well Cementing Co. v. Walker</i> , 329 U.S. 1 (1946)	23
<i>J.E.M. Ag Supply, Inc. v.</i> <i>Pioneer Hi-Bred Int’l, Inc.</i> , 534 U.S. 124 (2001)	12, 22
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007)	22
<i>Mayo Collaborative Servs. v.</i> <i>Prometheus Labs., Inc.</i> , 132 S. Ct. 1289 (2012)	<i>passim</i>

TABLE OF AUTHORITIES—continued

<i>Mazer v. Stein</i> , 347 U.S. 201 (1954).....	9
<i>Nautilus, Inc. v. Biosig Instruments, Inc.</i> , No. 13-369	24
<i>O’Reilly v. Morse</i> , 56 U.S. 62 (1853).....	31, 32
<i>Parker v. Flook</i> , 437 U.S. 584 (1978).....	18, 19, 28, 35
<i>Pearson v. Callahan</i> , 555 U.S. 223 (2009).....	20, 21
<i>Research Corp. Techs., Inc. v. Microsoft Corp.</i> , 627 F.3d 859 (Fed. Cir. 2010)	11
<i>SmartGene, Inc. v. Advanced Biological Labs., SA</i> , 2014 WL 259824 (Fed. Cir. 2014).....	29
<i>State St. Bank & Trust Co. v. Signature Fin. Grp. Inc.</i> , 149 F.3d 1368 (Fed. Cir. 1998)	11
<i>Tilghman v. Proctor</i> , 102 U.S. 707 (1880).....	16, 32
STATUTES	
17 U.S.C. § 117	9
35 U.S.C.	
§ 100.....	12
§ 101.....	<i>passim</i>
§ 102.....	22
§ 103.....	22
§ 112.....	22, 23, 24

TABLE OF AUTHORITIES—continued

OTHER AUTHORITIES

Robert D. Atkinson & Andrew S. McKay, <i>Digital Prosperity: Understanding the Economic Benefits of the Information Technology Revolution</i> (Info. Tech. & Innovation Found. 2007).....	7
BSA, <i>Patent Reform: The Verdict Is In</i> (2007)	6
Iain M. Cockburn & Megan MacGarvie, Nat'l Bureau of Econ. Research, Working Paper 13644 (2007)	10
The Boston Consulting Group, <i>The Great Software Transformation</i> (2013)	5
Henry J. Cittone, <i>Some Math Is Hard, Some Not: Rules for Patentable Subject Matter of Software</i> , 38 Rutgers Computer & Tech. L.J. 193 (2012).....	4
Julie E. Cohen & Mark A. Lemley, <i>Patent Scope and Innovation in the Software Industry</i> , 89 Cal. L. Rev. 1 (2001)	8, 20
Compact Oxford English Dictionary (2000)	12
Douglas Downing et. al, Dictionary of Computer & Internet Terms (11th ed. 2013).....	16
<i>Examination Guidelines for Determining Compliance with 35 U.S.C. § 112 and for Treatment of Related Issues in Patent Applications</i> , 76 Fed. Reg. 7162 (Feb. 9, 2011)	23

TABLE OF AUTHORITIES—continued

Robert W. Holleyman, BSA President and CEO, <i>Testimony before the United States House of Representatives Committee on En- ergy and Commerce, Subcommittee on Commerce, Manufacturing and Trade,</i> (Mar. 16, 2011)	6
Barry Jaruzelski et al., <i>The Global Innovation 1000: Navigating the Digital Future</i> , forth- coming 73 <i>Strategy & Business</i> 33 (Winter 2013)	6
David Kappos, Keynote Address at Center for American Progress, <i>An Examination of Software Patents</i> (Nov. 20, 2012)	4
Douglas Lippoldt & Piotr Stryszowski, Organ- isation for Economic Co-operation and De- velopment, <i>Innovation in the Software Sec- tor</i> (2009)	10
Bradford L. Smith & Susan O. Mann, <i>Innovation and Intellectual Property in the Software Industry: An Emerging Role for Patents?</i> , 71 <i>U. Chi. L. Rev.</i> 241 (2004).....	9
Nat'l Sci. Bd., <i>Science and Engineering Indica- tors</i> (2012).....	5
Report to Congressional Committees, <i>Intellectual Property: Assessing Factors That Affect Patent Infringement Litigation Could Help Improve Patent Quality</i> , GAO- 13-465	7

TABLE OF AUTHORITIES—continued

Ronald J. Mann & Thomas W. Sager, <i>Patents, Venture Capital, and Software Start-Ups</i> , 36 Research Policy 193 (2007)	9
U.S. Congress, Office of Technology Assessment, <i>Finding a Balance: Computer Software, Intellectual Property and the Challenge of Technological Change</i> 23 (1992).....	8
Webster's Third New International Dictionary (1986)	12

INTEREST OF THE *AMICUS CURIAE*

BSA | The Software Alliance is an association of the world's leading software and hardware technology companies. On behalf of its members, BSA promotes policies that foster innovation, growth, and a competitive marketplace for commercial software and related technologies. BSA members pursue patent protection for their intellectual property and as a group hold a large number of patents. They also create products that are frequently subject to unjustified patent infringement claims. Because patent policy is vitally important to promoting the innovation that has kept the United States at the forefront of software and hardware development, BSA members have a strong stake in the proper functioning of the U.S. patent system.¹

The members of the BSA include Adobe, Apple, ANSYS, Autodesk, AVG, Bentley Systems, CA Technologies, CNC/Mastercam, Dell, IBM, Intel, Intuit, McAfee, Microsoft, Minitab, Oracle, PTC, Rockwell Automation, Rosetta Stone, Siemens PLM, Symantec, Tekla, and The MathWorks.

SUMMARY OF ARGUMENT

Software innovations are a critical engine of today's economy that touch myriad aspects of everyday life.

¹ Pursuant to Rule 37.6, *amicus* affirms that no counsel for a party authored this brief in whole or in part and that no person other than *amicus* and its counsel made a monetary contribution to its preparation or submission. The parties' letters consenting to the filing of this brief have been filed with the Clerk's office.

The discussion of software patentability, however, is often dominated by a significantly different kind of patent claim. Like the claims in this case, some patents merely take a well understood, fundamental concept (here, the business method of intermediated settlement) and claim application of that idea on a computer. That amounts to nothing more than a patent on the idea itself. Section 101 bars such claims, which are not “software” at all.

Concluding that the claims are not patentable in this straightforward case—which is effectively “*Bilski* on a computer”—fully accords with the principle that true software innovations are patent-eligible.

First, software engineering is like any other engineering discipline: innovations are the product of hard work and significant investment, and they add substantial value to the economy. This point is proven by the tens of billions of dollars that companies spend each year researching and developing new software innovations. The benefits of these advances are seen in all aspects of the economy, through pathbreaking new products that touch virtually every aspect of modern life. Given the central importance of software to the economy, patents for the software innovations that power the Information Age are every bit as justified as the patents for industrial processes that powered American growth a century ago.

Second, software innovations generally are patentable under Section 101. Software patents describe a “process” for accomplishing a particular end, which is an express statutory basis for patent eligibility. And *Bilski v. Kappos*, 130 S. Ct. 3218, 3224 (2010), expressly rejected artificial limitations on

what qualifies as a patent-eligible “process.” Moreover, software innovations create a new “machine” when they program a computer to perform a particular, useful function. Finally, software innovations are not categorically exempt from Section 101 as laws of nature, physical phenomenon, or abstract ideas.

Third, because the Patent Act contains very substantial additional requirements before a patent may be granted, Section 101 is not—nor should be viewed as—the sole guardian of patent quality. Section 101 applies to software claims in the same manner as to innovations in every other field of endeavor. Most software claims—which generally relate to needs, problems, or opportunities unique to the digital environment—do not implement a law of nature, physical phenomenon, or abstract idea at all. With respect to these claims, the implicit exemptions to Section 101 are irrelevant.

When a claimed invention *does* implement such a concept, a court must consider “[w]hat else is there in the claims,” to determine whether it does “significantly more” than describe the unpatentable concept. *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1297 (2012). If it does not do “significantly more,” the claim is barred by Section 101.

Fourth, application of these principles is straightforward in this case. Unlike most true software claims, the claims here *do* implement an abstract idea—the idea of intermediated settlement. And these claims do not do “significantly more” than describe the abstract idea as implemented on a computer. The various forms of the claims do not alter the conclusion that they are not patentable under Section 101.

ARGUMENT

I. Society Has Reaped Enormous Benefits From Computer Software Patents.

“[M]any breathtaking software-implemented innovations power our modern world, at levels of efficiency and performance unthinkable even just a few years ago,” then-PTO Director David Kappos recently explained. David Kappos, Keynote Address at Center for American Progress, *An Examination of Software Patents* (Nov. 20, 2012), <http://tiny.cc/33zfow>. “[P]atent protection” therefore “is every bit as well-deserved for software-implemented innovation” as for earlier innovations “that enabled man to fly, and before that for the innovations that enabled man to light the dark with electricity, and before that for the innovations that enabled the industrial revolution.” *Ibid.*

The patentability of software is critically important to the U.S. economy. Subjecting software to patentability rules more restrictive than those applicable to other categories of inventions would reduce economic growth and diminish innovation in software that touches virtually every aspect of American life.

A. Software Contributes Substantially To The U.S. Economy.

Just as civil engineers design the physical infrastructure of bridges and roads, software engineers build the digital world that touches nearly every aspect of modern life. Indeed, “most of the planet is currently run by software” as “[o]ur financial systems, energy production, transportation networks and a host of other fundamental systems are run using software.” Henry J. Cittance, *Some Math Is Hard*,

Some Not: Rules for Patentable Subject Matter of Software, 38 Rutgers Computer & Tech. L.J. 193, 193-194 (2012).

Software is used by “[p]atients with chronic diseases [who] wear[] devices that monitor and help to manage their conditions;” by “cities [to] monitor, manage, and reroute traffic during peak times;” and by insurers to “adjust[] premiums based on real-time driving habits of customers who agree to have sensors placed in their cars.” The Boston Consulting Group (“BCG”), *The Great Software Transformation* 11-12 (2013), <http://tiny.cc/rkd99w>.

Software controls everyday devices, like tablet computers and smartphones—of which there will be more than 5 billion by 2018. BCG, *supra*, at 6. It is “hard to overestimate the pivotal role of software in altering a wide range of traditional ways of working;” indeed, “[s]oftware is the engine that has driven many, if not most, of the most disruptive business models introduced over the past 25 years.” *Id.* at 5. In short, “software is at the heart of the global economy.” *Ibid.*

Investment in software reflects its critical importance to American industry. In 2008, companies invested approximately \$46.9 billion in research and development for software and computer-related services—approximately 16% of *total* industrial R&D expenditures for the nation. Nat’l Sci. Bd., Science and Engineering Indicators, at 4-21 & 4-23 (2012), <http://tiny.cc/xicwbx>. Reflecting this substantial investment, software firms are leading innovators, “with 77% of companies” engaged in software development “reporting the introduction of a new product or service compared to the 7% average for all non-manufacturing industries.” *Id.* at 6-47.

Software R&D spending is growing at substantial rates. In 2013, R&D spending in the software and Internet sector grew approximately 22.1% from the prior year, compared to just 5.8% R&D growth for the whole economy. Barry Jaruzelski et al., *The Global Innovation 1000: Navigating the Digital Future*, forthcoming 73 *Strategy & Business* 33, 36-37 (Winter 2013), <http://tiny.cc/2smeax>. BSA member companies spend in excess of \$32 billion each year on research and development to expand their innovation portfolios. See BSA, *Patent Reform: The Verdict Is In* 4 (2007), <http://tiny.cc/kjd99w>.

These substantial investments in software R&D confirm that innovation in software is just like innovation in any other engineering discipline—it takes substantial amounts of ingenuity, effort, and expense to develop new technologies.

Software is also a critical driver of investment in the United States. Software companies accounted for \$21.6 billion of foreign direct investment in the United States in 2009. Nat'l Sci. Bd., *supra*, at 6-46, tbl. 6-7. And venture capital firms invested approximately \$18 billion between 2007 and 2010 in software companies. *Id.* at 6-58, to -60 & fig. 6-51.

Moreover, the software industry creates a substantial number of high-paying American jobs. Currently, software companies and related services employ approximately two million U.S. workers, paying salaries that are roughly 195% of the national average. Robert W. Holleyman, BSA President and CEO, *Testimony before the United States House of Representatives Committee on Energy and Commerce, Subcommittee on Commerce, Manufacturing and Trade*, at 2 (Mar. 16, 2011), <http://tiny.cc/p3nlow>.

Software sales outside the United States constitute a substantial portion of U.S. export markets, significantly strengthening the U.S. economy. Software accounts for approximately \$36 billion of U.S. exports, and leading software companies make as much as 60% of their revenue on overseas sales. Holleyman, *supra*, at 2.

The benefits of software are not limited to the technology sector itself. Information technology innovations have been described as “the key factor responsible for reversing the 20-year productivity slowdown from the mid-1970s to the mid-1990s and in driving today’s robust productivity growth.” Robert D. Atkinson & Andrew S. McKay, *Digital Prosperity: Understanding the Economic Benefits of the Information Technology Revolution* 10 (Info. Tech. & Innovation Found. 2007), <http://tiny.cc/3ld99w>. Software helps companies “collaborate more effectively internally and externally, scale operations faster, operate more efficiently, and innovate and experiment more strategically.” BCG, *supra*, at 11. Today, “all companies are effectively tech companies.” *Ibid.*

Given the integral role computer software plays in all aspects of modern life, it comes as little surprise that inventors are increasingly seeking patent protection for software innovations; “the number of software-related patents grew as computers were integrated into a greater expanse of everyday products.” See U.S. Gov’t Accountability Office, Report to Congressional Committees, *Intellectual Property: Assessing Factors That Affect Patent Infringement Litigation Could Help Improve Patent Quality*, GAO-13-465, at 11-13 (2013), <http://tiny.cc/0md99w>. Currently, about half of all new patents relate in some manner to computer software. *Ibid.*

B. The Continued Patentability Of Software Is Critical To Economic Growth.

The availability of patent protection for software inventions has been an essential incentive for the innovation that has produced this wide array of benefits. New limitations on that patent protection would therefore inflict very significant injury on the U.S. economy.

Simple economics makes clear that, if patent protection for software were curtailed, the adverse consequences would be swift and severe. “Discrimination against a form of innovation that is increasingly critical to technological advancement, indeed that in many areas dominates technological advancement, makes no sense.” Kappos, *supra*. See also Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Cal. L. Rev. 1, 5 (2001) (“[B]oth economic theory and practical experience suggest that the availability of patents for software promotes innovation by supplying (additional) incentives to inventors.”).

Indeed, as early as 1992, congressional reports recognized that “patent protection is of importance to the U.S. software industry, both domestically and in the global market.” U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property and the Challenge of Technological Change* 23 (1992), <http://tiny.cc/gkewbx>.

First, because patent protection is a critical incentive to expenditures for software research and development, limitation of software patentability would lead to a decline in software innovations. Without in-

tellectual property protection, prospective software entrepreneurs face serious risks that competitors will free-ride on their innovations by pilfering the essential elements of a software program. See, e.g., Bradford L. Smith & Susan O. Mann, *Innovation and Intellectual Property in the Software Industry: An Emerging Role for Patents?*, 71 U. Chi. L. Rev. 241, 241-242 (2004). With proper protection, by contrast, potential innovators are motivated to pursue new inventions and to proceed to commercial development to collect their economic rewards. *Id.* at 256-257.²

Second, any new obstacles to software development would carry a penalizing multiplier effect that could threaten the broader economy. Because software is at the heart of the efficiency revolution that has contributed vastly to U.S. economic growth over the past few decades, causing companies to divert their resources away from software research and development would have a ripple effect on productivity, affecting all segments of the economy. See BCG, *supra*, at 11.

Third, limiting software patentability would particularly harm small and start-up entities, which rely upon software patents in order to gain critical early funding. Software patents “play a role of some importance in the development of firms seeking to enter the software industry” insofar as they significantly improve a company’s efforts to obtain venture capital. Ronald J. Mann & Thomas W. Sager, *Patents*,

² Although software is protected by copyright (see 17 U.S.C. § 117), “[u]nlike a patent, a copyright gives no exclusive right to the art disclosed; protection is given only to the expression of the idea—not the idea itself.” *Mazer v. Stein*, 347 U.S. 201, 217 (1954).

Venture Capital, and Software Start-Ups, 36 Research Policy 193, 194 (2007), <http://tiny.cc/snd99w>. Thus, in the software space, start-up “[f]irms that have higher numbers of patents and patent applications pending are more likely to receive funding from outside investors, and more likely to subsequently ‘exit’ from the entrepreneurial phase through IPO or acquisition.” Iain M. Cockburn & Megan MacGarvie, Nat’l Bureau of Econ. Research, Working Paper 13644, at 42 (2007), <http://tiny.cc/nod99w>.

Fourth, any undue limits on software patents would cause firms to move their software development efforts outside the United States. “[A] worldwide comparison of patents issued for software-related inventions” shows that “firms will locate their R&D activities in nations with strong intellectual property rights and sound legal institutions.” Douglas Lippoldt & Piotr Stryszowski, Organisation for Economic Co-operation and Development, *Innovation in the Software Sector* 14-15 (2009), <http://tiny.cc/usd99w>. Given that software is a critical aspect of continued U.S. economic success, there can be no justification for diminishing patent protection and driving this enormously important industry outside the country.

II. Software Innovations Are Eligible For Patent Protection.

Section 101’s broad language—extending patent eligibility to “any * * * process, machine, manufacture, or composition of matter, or any new and useful improvement thereof”—plainly encompasses software innovations. Indeed, while the *en banc* Federal Circuit disagreed about many issues in this case, not one judge disputed that software, as a general matter, is patent-eligible.

That unanimity is not surprising. Decades ago, this Court found claims involving the use of a “computer program” eligible for patent protection. *Diamond v. Diehr*, 450 U.S. 175, 181 (1981). A claim “does not become nonstatutory simply because it uses a * * * computer program.” *Id.* at 187. The Federal Circuit has found software patentable in a substantial number of cases.³

Most patent applications involving software satisfy Section 101: software innovations are eligible under the plain terms of the statute, and the implicit exemptions to Section 101 do not categorically exclude software.

A. Software Is Patent-Eligible As Both A “Process” And As A “Machine.”

This Court has recognized that Section 101 should be interpreted to extend patent eligibility broadly. Because the statutory provision contains “expansive terms modified by the comprehensive ‘any,’ Congress plainly contemplated that the patent laws would be given wide scope,” so as “to ensure that ingenuity should receive a liberal encouragement.” *Bilski*, 130 S. Ct. at 3225 (quotations omitted). “Section 101 is a ‘dynamic provision designed to encompass new and unforeseen inventions.’” *Ibid.*

³ See, e.g., *Research Corp. Techs., Inc. v. Microsoft Corp.*, 627 F.3d 859, 862-864 (Fed. Cir. 2010); *Eolas Techs. Inc. v. Microsoft Corp.*, 399 F.3d 1325 (Fed. Cir. 2005); *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) (en banc). So too had its predecessor court, the United States Court of Customs and Patent Appeals. See, e.g., *In re Freeman*, 573 F.2d 1237 (C.C.P.A. 1978); *In re Bernhart*, 417 F.2d 1395, 1396 (C.C.P.A. 1969).

(quoting *J.E.M. Ag Supply, Inc. v. Pioneer Hi-Bred Int'l, Inc.*, 534 U.S. 124, 135 (2001)).

Software qualifies as a “process” under Section 101, which the Patent Act defines as a “process, art or method.” 35 U.S.C. § 100(b). The Court looks to “dictionary definitions” (*Bilski*, 130 S. Ct. at 3226) to determine the “common meaning” of Section 101’s terms (*Diehr*, 450 U.S. at 182).

A process is “a particular method or system of doing something, producing something, or accomplishing a specific goal.” Webster’s Third New International Dictionary 1808 (1986). A “method” is “a way, technique, or process of or for doing something.” *Id.* at 1423.

Software innovations fall well within this broad definition of a process. By its very nature, a software innovation specifies a series of actions or operations to be taken by computer hardware in order to accomplish some goal. Likewise, software innovations will disclose a systematic procedure or technique that is used in computer hardware. In fact, “software” itself is defined as “[t]he programs and procedures required to enable a computer to perform a specific task.” The Compact Oxford English Dictionary 1819 (2000).

There is no basis for imposing an artificial limitation on the term “process” that would exclude software. The Court “has ‘more than once cautioned that courts should not read into the patent laws limitations and conditions which the legislature has not expressed.’” *Bilski*, 130 S. Ct. at 3226 (quoting *Diehr*, 450 U.S. at 182). This is so, in no small part, because any such artificial, “categorical rule[]” “might have wide-ranging and unforeseen impacts.” *Id.* at 3229.

Moreover, “Congress employed broad general language in drafting § 101 precisely because * * * inventions are often unforeseeable.” *Diamond v. Chakrabarty*, 447 U.S. 303, 316 (1980).

In *Bilski*, for example, the Court expressly rejected the contention that a process is eligible for patent protection only when it passes the so-called “machine-or-transformation” test. That approach would have limited patent-eligible processes to only those that are “tied to a particular machine or apparatus” or that “transform[] * * * [an] article into a different state or thing.” *Bilski*, 130 S. Ct. at 3225 (quotation omitted). Such a restrictive approach, the Court stated, would be “inconsistent with the text and the statute’s purpose and design.” *Id.* at 3226.

While “[t]he machine-or-transformation test may well provide a sufficient basis for evaluating processes similar to those in the Industrial Age,” because such inventions are often “grounded in a physical or other tangible form,” “there are reasons to doubt whether the test should be the sole criterion for determining the patentability of inventions in the Information Age.” *Bilski*, 130 S. Ct. at 3227 (plurality opinion). See also *id.* at 3228 (rejecting argument that “business methods are categorically outside of § 101’s scope”).

Software is also patent-eligible as a “new and useful improvement” of a “machine.” 35 U.S.C. § 101. A software algorithm “creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software.” *In re Alappat*, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc). In fact, because “a software process is often interchangeable with a

hardware circuit,” “the line of demarcation between a dedicated circuit and a computer algorithm accomplishing the identical task is frequently blurred and is becoming increasingly so as the technology develops.” *Id.* at 1583 (Rader, J., concurring). Software in application creates a new improvement of an existing machine—precisely what Section 101 requires.

The Patent Act is specifically designed to encompass inventions not imagined at the time of its enactment. Denying patent protection for “inventions in areas not contemplated by Congress * * * would frustrate the purposes of the patent law,” because “[a] rule that unanticipated inventions are without protection would conflict with the core concept of the patent law that anticipation undermines patentability.” *Chakrabarty*, 447 U.S. at 315-316.

B. Software Innovations Are Not Categorically Exempt By The Implied Exceptions To Section 101.

This Court has identified “three specific exceptions to § 101’s broad patent-eligibility principles: laws of nature, physical phenomena, and abstract ideas.” *Bilski*, 130 S. Ct. at 3225. These are “the basic tools of scientific and technological work” and “monopolization of those tools through the grant of a patent might tend to impede innovation more than it would tend to promote it.” *Mayo*, 132 S. Ct. at 1293 (quoting *Gottschalk v. Benson*, 409 U.S. 63, 67 (1972)).

The Court has warned that “too broad an interpretation” of these implicit exemptions to Section 101 “could eviscerate patent law” because “all inventions at some level embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract

ideas.” *Mayo*, 132 S. Ct. at 1293. Thus, “[w]hile a scientific truth, or the mathematical expression of it, is not a patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be.” *Id.* at 1294 (quoting *Diehr*, 450 U.S. at 188).

There is no serious argument that all software is itself a law of nature or a physical phenomenon. Nor is software inherently an “abstract idea.”

The Court held in *Bilski* that an “abstract idea” is a “fundamental * * * practice long prevalent.” 130 S. Ct. at 3231 (quotation omitted). It determined that the concept of hedging risk, which is “taught in any introductory finance class,” falls within that category. *Ibid.* (quotation omitted). Permitting a patent for that concept would “pre-empt use of this approach in all fields,” and “would effectively grant a monopoly over an abstract idea.” *Ibid.*

Software is a recent innovation tied to the development of computers, and therefore cannot be categorized as a “fundamental practice long prevalent,” akin to risk hedging. And the investment of hundreds of billions of dollars in software development demonstrates why numerous software innovations satisfy Section 101’s requirement that a claimed invention be “new and useful.” *Bilski*, 130 S. Ct. at 3225.

Those who argue that software should not be patentable contend that all software is an “algorithm,” and that all “algorithms” are ineligible for patent protection. That contention is plainly wrong. Many “algorithms” are patent eligible; in fact, *all* process patents qualify as algorithms.

An “algorithm” is “a sequence of instructions that tells how to solve a particular problem,” which is “specified exactly” and contains “a finite number of steps.” Douglas Downing et. al, Dictionary of Computer & Internet Terms (11th ed. 2013). “An algorithm might be written in a computer program, or it can be a set of instructions for a person to follow.” *Ibid.* It can be described in words or represented graphically, such as through a flow chart.

Thus, *any* process with finite and definite steps is an algorithm, or—at the very least—can be written as an algorithm. A parent who posts on the kitchen refrigerator—“If kitchen trash can is full, take to the garage. If garage trash can is full, take to the curb.”—has written an algorithm. An “algorithm” is not limited to a mathematical formula; any specified sequence of steps qualifies.

For that reason, virtually every patent-eligible process discloses an algorithm. In *Diehr*, for example, the “claims describe in detail a step-by-step method” for curing rubber; in other words, an algorithm. 450 U.S. at 184. And the *Diehr* Court emphasized that “[i]ndustrial processes such as this are the types which have historically been eligible to receive the protection of our patent laws.” *Ibid.* See also *Tilghman v. Proctor*, 102 U.S. 707, 722 (1880) (“A manufacturing process is clearly an art, within the meaning of the law.”).

Of course, an algorithm *could* implement an “abstract idea.” One might write an algorithm, for example, describing the risk hedging at issue in *Bilski* as a series of generalized steps. Likewise, one could write an algorithm disclosing the steps to dose certain medications by measuring metabolites in the blood, a law of nature, as at issue in *Mayo*.

But the fact that an algorithm *can* implement an abstract idea does not mean that *all* algorithms are abstract ideas and thus patent ineligible—any more than the fact that a chemical formula can be used to describe a substance found in nature means that all chemical formulae are ineligible under Section 101. Otherwise, the category of patent-eligible “processes” would be rendered a null set—a result fundamentally at odds with the text of Section 101 and centuries of practice.

Previous uses of the term “algorithm” by this Court confirm the conclusion that only those algorithms describing an abstract idea or law of nature are nonpatentable.

Benson involved a patent that claimed the process of converting binary-coded decimal (BCD) signals into pure binary form. 409 U.S. at 65. The Court viewed this process, which was the application of a “mathematical formula,” as “so abstract and sweeping as to cover both known and unknown uses of the BCD to pure binary conversion.” *Id.* at 68, 71. The formula could be used in applications that “vary from the operation of a train to verification of drivers’ licenses to researching the law books for precedents.” *Id.* at 68. It was the extraordinarily abstract nature of the claimed process, and the lack of any limitation to a concrete application, that led the Court to reject the claim.

Thus, in finding the claim ineligible the Court explained that “the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.” *Benson*, 409 U.S. at 72. That is, the *specific* algorithm at issue in *Benson*—which was defined there as the

mathematical formula—was patent ineligible. *Benson* did not hold *all* algorithms patent ineligible.

Next, in *Parker v. Flook*, 437 U.S. 584, 585 (1978), the Court considered alarm limits used during catalytic conversions. The process had three steps: measuring certain values (like temperature), calculating these values via a claimed mathematical formula, and updating the alarm-limit that would signal danger or inefficiency. *Id.* at 586. The mathematical formula at issue was “a law of nature” (*id.* at 589), the Court said, because it “reveal[ed] a relationship that has always existed.” *Id.* at 593 n.15.

The Court concluded that the “process is unpatentable under § 101, not because it contains a *mathematical* algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention.” *Flook*, 437 U.S. at 594 (emphasis added). The Court’s holding that the claim was not patentable, accordingly, was based on its conclusion that the particular algorithm was nothing more than the mathematical formula itself.

In *Diehr*, the Court again referred to a mathematical formula, there the Arrhenius equation, as an algorithm. 450 U.S. at 186. Once more, the *particular* algorithm at issue described a “law of nature”—an equation that corresponded to an observable phenomenon. *Ibid.*

Diehr expressly noted that “[t]he term ‘algorithm’ is subject to a variety of definitions,” including the “significantly broader” definition of “[a] fixed step-by-step procedure for accomplishing a given result.” 450 U.S. at 186 n.9. The Court emphasized that its earlier decisions in *Benson* and *Flook*, “regarding the

patentability of ‘algorithms’ are necessarily limited to the more narrow definition employed by the Court”—a mathematical formula describing an abstract idea. *Ibid.* The Court, therefore, did “not pass judgment on whether processes falling outside the definition previously used by this Court,” but within the broader definition of the term, “would be patentable subject matter.” *Ibid.*

Benson, *Flook*, and *Diehr*, accordingly, establish the unexceptional principle that a particular “algorithm” that simply relays a law of nature or an abstract idea is patent-ineligible. But that has no bearing on the patent-eligibility of an “algorithm” that is *not* a law of nature or abstract idea. Rather, when the term “algorithm” is used in the broader sense to describe a step-by-step process of accomplishing a particular result, it is plainly within the scope of Section 101; otherwise every process patent would be invalid.

When software is described categorically as an “algorithm,” it is in that broader sense, because software specifies a step-by-step process by which a computer arrives at a particular result. For that reason, software algorithms—which embody a process like any other created by human engineering—are not categorically excluded from Section 101.

C. *Stare Decisis* Precludes A Categorical Bar On Software Patentability.

This Court, the Federal Circuit, and the Federal Circuit’s predecessor have consistently held software to be patent eligible. See page 11 & n.3, *supra*. *Stare decisis* therefore precludes any diminution in patent protection for software.

The PTO has issued hundreds of thousands of software patents. Indeed, thirteen years ago, noting that “[w]ith some eighty thousand software patents already issued,” leading commentators concluded that “software patentability is a matter for the history books.” Cohen & Lemley, 89 Cal. L. Rev. at 4.

Stare decisis has special force in this context because “[f]undamental alterations” in intellectual property “rules risk destroying the legitimate expectations of inventors in their property,” and courts therefore “must be cautious before adopting changes that disrupt the settled expectations of the inventing community.” *Festo Corp. v. Kinzoku Kogyo Kabushiki Co.*, 535 U.S. 722, 739 (2002). “Considerations in favor of *stare decisis* are at their acme in cases involving property and contract rights, where reliance interests are involved.” *Pearson v. Callahan*, 555 U.S. 223, 233 (2009) (quotation omitted).

The settled expectations of a massive industry, expectations that drive tens of billions of dollars worth of research and development spending each year and support millions of jobs, therefore preclude a decision limiting the patentability of software innovations.

III. Section 101 Serves An Important, But Limited, Screening Function.

Section 101’s threshold test is important, but it is not the only legal requirement that a claimed invention must satisfy in order to obtain a patent. Other significant standards play a critical role in guaranteeing patent quality.

The Section 101 test applies to software claims in the same manner the Court has applied it to patent claims in other fields of invention. If the software

claim does not involve a law of nature, physical phenomenon, or abstract idea—and the vast majority of software claims do not—Section 101 is satisfied. If the claim does involve such a concept, the next step is to determine whether the claim does “significantly more” than merely describe that law, phenomenon, or idea. *Mayo*, 132 S. Ct. at 1297.⁴

A. Section 101 Need Not And Should Not Incorporate Other Patent Act Requirements.

The content of Section 101’s “threshold test” for patent eligibility (*Bilski*, 130 S. Ct. at 3225) is informed by the other significant statutory requirements that a patent applicant must satisfy—and that therefore protect against unjustified patents. Congress’s decision to include these requirements makes clear that it did not intend Section 101’s “threshold test” to perform all—or even most—of the

⁴ Some *amici* urge the Court to adopt a rule *requiring* a district court to address Section 101 issues at the outset of the litigation; others seek a rule *requiring* a court to await claim construction. Compare, *e.g.*, Retailers Br. 6-10 with Ass’n of Bar of New York Br. 19-24. Neither approach is correct, as “the judges of the district courts * * * are in the best position to determine the order of decisionmaking that will best facilitate the fair and efficient disposition of each case.” *Pearson*, 555 U.S. at 242.

The Court’s reference to Section 101 as a “threshold test” (*Bilski*, 130 S. Ct. at 3225) does not command *when* a court must conduct that analysis. In fact, in *Dann v. Johnston*, 425 U.S. 219, 220 (1976), the Court held so expressly: while the parties “presented lengthy arguments addressed to the question of the general patentability of computer programs,” the Court concluded that it had “no need to treat that question in this case” because “in any event respondent’s system is unpatentable on grounds of obviousness.”

job of precluding patent grants to ineligible claims. Although these other “inquir[ies] might sometimes overlap” with the Section 101 analysis (*Mayo*, 132 S. Ct. at 1304), “[t]hese limitations serve a critical role in adjusting the tension, ever present in patent law, between stimulating innovation by protecting inventors and impeding progress by granting patents when not justified by the statutory design” (*Bilski*, 130 S. Ct. at 3229).

First, a patent claim must describe an invention that is “novel.” 35 U.S.C. § 102(a). A claim is invalid if it was “patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention.” *Ibid.* See also *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 150 (1989).

Second, the claimed invention must be nonobvious. 35 U.S.C. § 103. A claim is not patentable if “the differences between the claimed invention and the prior art” “would have been obvious * * * to a person having ordinary skill in the art to which the claimed invention pertains.” *Ibid.* See also *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007).

Third, the patent must fully and particularly describe the claimed invention. 35 U.S.C. § 112. Disclosing to the world one’s innovation “is the *quid pro quo* of the right to exclude.” *J.E.M. Ag Supply*, 534 U.S. at 142 (quotation omitted).

This requires an inventor to provide “a written description of the invention” that would “enable any person skilled in the art to which it pertains * * * to make and use the same.” 35 U.S.C. § 112(a). The description must also “set forth the best mode contem-

plated by the inventor * * * of carrying out the invention.” *Ibid.* The claims must be sufficiently definite; they must “particularly point[] out and distinctly claim[] the subject matter which the inventor or a joint inventor regards as the invention.” *Id.* § 112(b). And an inventor cannot merely claim the purported functions of an invention; the patent must disclose “a means or step for performing a specified function.” *Id.* § 112(f). See also *Halliburton Oil Well Cementing Co. v. Walker*, 329 U.S. 1, 12 (1946) (noting “the broadness, ambiguity, and overhanging threat of the functional claim”).

The PTO recently issued formal regulations that underscore the importance of Section 112. See *Supplementary Examination Guidelines for Determining Compliance with 35 U.S.C. § 112 and for Treatment of Related Issues in Patent Applications*, 76 Fed. Reg. 7162 (Feb. 9, 2011). The guidance emphasizes that claims must be sufficiently definite (*id.* at 7163-7167), and—particularly with respect to software patents—that claims may not assert a function without disclosing the means of accomplishing that function (*id.* at 7167-7168). In the context of software, to satisfy Section 112, a patent generally must disclose a sufficiently specific algorithm to accomplish the claimed function. *Id.* at 7168. A year after issuing this guidance, the PTO reported that “examiners were making 4 to 6 percent more rejections in patent applications across all technology areas” based on Section 112. GAO, *Intellectual Property*, *supra*, at 40.

Section 112 has proven an effective mechanism for invalidating low-quality software patents that do not adequately disclose a means for implementing the claimed function. See, *e.g.*, *Blackboard, Inc. v. Desire2Learn, Inc.*, 574 F.3d 1371, 1385 (Fed. Cir.

2009); *Aristocrat Techs. Australia Pty Ltd. v. Int'l Game Tech.*, 521 F.3d 1328, 1333 (Fed. Cir. 2008); *Finisar Corp. v. DirecTV Grp., Inc.*, 523 F.3d 1323, 1340-1241 (Fed. Cir. 2008) (“Simply reciting ‘software’ without providing some detail about the means to accomplish the function is not enough.”).

Section 112, accordingly, weeds out overbroad, ambiguous, or un-enabled claims, or claims that fail to set forth the best mode or that fail to disclose a means to accomplish a claimed function.⁵

These three standards, with their carefully-calibrated standards grounded in express statutory language, provide significant protections against unjustified patent grants.

B. Section 101’s Standards Apply To Software Claims In The Same Manner As Claimed Inventions In Other Fields Of Endeavor.

A *particular* claimed software invention—like a particular claimed invention of any other type—may be ineligible under Section 101. *Cf. Ass’n for Molecular Pathology v. Myriad Genetics, Inc.*, 133 S. Ct. 2107, 2119 (2013) (holding “cDNA sequence” which “is not naturally occurring” is patent eligible, while a naturally-occurring DNA sequence is *not* patent eligible).

Section 101’s implicit exclusions apply to software claims in the same manner that the Court has applied them to other innovations. The Court’s Sec-

⁵ The Court is well aware of the importance of Section 112, having recently granted certiorari in *Nautilus, Inc. v. Biosig Instruments, Inc.*, No. 13-369.

tion 101 jurisprudence provides two critical guideposts.

First, a significant majority of software claims are unrelated to a law of nature, physical phenomenon, or abstract idea. These claims are eligible for patent protection under Section 101 without further inquiry.

Second, a software claim that *does* implement a law of nature, physical phenomenon, or abstract idea is eligible for patent protection if it does “significantly more” than, in general terms, claim the application of that concept on a computer. But the prohibition on patenting a law of nature or abstract idea cannot be circumvented by merely claiming to apply that law or idea “on a computer”—such claims are excluded by Section 101.

1. *A Software Algorithm Unrelated To A Law Of Nature, Physical Phenomenon, Or Abstract Idea Is Necessarily Eligible For Patent Protection Under Section 101.*

The software patent claims in this case—like those that have received the lion’s share of judicial attention—are *atypical* of software innovations generally. Most software claims do not merely implement on a computer an abstract concept or law of nature that pre-dated computers or exists apart from them. That is, unlike this case, most claims do not simply claim “an abstract idea implemented on a computer.” The typical software claim instead involves innovations that are unique to the digital world itself.

Following are only a few examples of the tens of thousands of such software inventions.

Computer functionality:

- *Lempel-Ziv-Welch data compression*: A lossless data compression software that reduces a text file to about half its original size, which was critical for the development of computer networking. The patent discloses very specific algorithms controlling the manipulation of digital data. U.S. Pat. Nos. 4,464,650 & 4,814,746.
- *RSA security token*: A cryptology software program that provides secure online communications. These claims also disclosed substantially complex and specific algorithms governing the claimed process. U.S. Pat. No. 4,405,829.
- *Philips multiplexing*: A software system that permits more efficient transmission of digital video content; it is broadly used in a variety of applications, including online videos and DVDs. This software makes possible streaming of online content. U.S. Pat. No. 5,333,135.

Telecommunications:

- *Qualcomm CDMA technology*: Code Division Multiple Access is used by most U.S. cell phones to allow multiple users to share the same frequencies. This technology is critical to efficient cellular networks. U.S. Pat. Nos. 5,103,459, 5,101,501, 5,109,390, & 5,056,109.

Digital Imagery:

- *Adobe Photoshop's "healing brush" feature*: A software process that permits a user to quickly manipulate a digital photo by remov-

ing unwanted blemishes. U.S. Pat. No. 6,587,592.

Entertainment:

- *10NES*: A software system designed by Nintendo that ensures only authentic, authorized games may be played on a Nintendo game console. U.S. Pat. No. 4,799,635.
- *Microsoft Kinect gesture recognition and skeletal tracking*: Software that creates a means for tracking human gestures as an input mechanism for computers. In addition to entertainment, this technology provides assistive technologies for people with disabilities, creates more engaging educational tools for children, and helps doctors during surgeries. U.S. Pat. Nos. 7,974,443 and 7,996,793.

None of these claims implements an abstract idea, law of nature, or physical phenomenon; instead, they are tethered to innovations specific to the digital world. In these circumstances—which describe a huge percentage of software patents—no further analysis under Section 101 is required to determine that the claim is patent-*eligible*.

2. *A Software Claim That Does Implement A Law Of Nature, Physical Phenomenon, Or Abstract Idea Is Patent Eligible If It Does “Significantly More” Than Describe The Patent-Ineligible Concept.*

Some software claims, however, *do* implement laws of nature, physical phenomena, or abstract ideas via a computer. When such a software claim does “significantly more” than claim all computer implementation of that concept, the claim is eligible for

patent protection. *Mayo*, 132 S. Ct. at 1297. “[A] process is not unpatentable simply because it contains a law of nature” (*Flook*, 437 U.S. at 590); an “*application* of” an “unpatentable” concept may “transform” it into something patent-eligible. *Mayo*, 132 S. Ct. at 1294.

Diehr illustrates this principle at work. There, the claim used, in part, Arrhenius’ equation to determine the length of time rubber products should remain in curing molds. 450 U.S. at 187-188. The claim would not “pre-empt the use of that equation” in other contexts; it only would “foreclose from others the use of that equation in conjunction with all of the other steps in their claimed process.” *Id.* at 187. And there were several significant steps other than applying the equation. *Ibid.*

Accordingly, while “Arrhenius’ equation is not patentable in isolation,” “when a process for curing rubber is devised which incorporates in it a more efficient solution of the equation, that process is at the very least not barred at the threshold by § 101.” *Diehr*, 450 U.S. at 188. The process, which in part involved a computer innovation, was patentable because it applied the Arrhenius equation to solve a real, practical, and limited problem.

If, on the other hand, the patent application simply tacks an extremely general computer limitation to an abstract idea or law of nature—without anything more—it is not patent-eligible. One cannot take a concept that exists outside of the digital environment and that, standing alone, would not qualify as patent eligible subject matter, and transform it into a patentable eligible innovation simply by claiming generally to “apply it” via a computer. *Cf. Mayo*, 132 S. Ct. at 1294.

As this Court made clear, “Einstein could not patent his celebrated law that $E=mc^2$.” *Mayo*, 132 S. Ct. at 1293. Nor could he “have patented his famous law by claiming a process consisting of simply telling linear accelerator operators to refer to the law to determine how much energy an amount of mass has produced.” *Id.* at 1297. Nor can that formula be patented by directing one to “compute it.”

That conclusion results from the principle that meaningless limitations cannot transform a law of nature, physical phenomenon, or abstract idea into something that is patent eligible. Such meaningless limitations include specifying a “particular technological environment” (*Bilski*, 130 S. Ct. at 3230), adding “insignificant postsolution activity” (*ibid.*), or describing “well-understood, routine, [or] conventional activity” (*Mayo*, 132 S. Ct. at 1294).

A mere limitation to “apply” a patent-ineligible concept “on a computer,” is a field-of-use limitation, postsolution activity, *and* conventional. Thus, claiming in generalized terms a computer limitation does not, by itself, transform a patent-ineligible concept into something that qualifies under Section 101.⁶

⁶ This is the approach that the plurality opinion took below. Pet. App. 28a. And the Federal Circuit has often concluded that software patent claims are excluded by Section 101 when they couple a generalized computer-implementation limitation with an abstract idea. See, e.g., *SmartGene, Inc. v. Advanced Biological Labs., SA*, 2014 WL 259824 (Fed. Cir. 2014); *Accenture Global Servs., GmbH v. Guidewire Software, Inc.*, 728 F.3d 1336, 1344 (Fed. Cir. 2013); *Bancorp Servs., L.L.C. v. Sun Life Assur. Co.*, 687 F.3d 1266, 1278 (Fed. Cir. 2012); *Fort Props., Inc. v. Am. Master Lease LLC*, 671 F.3d 1317, 1323-1324 (Fed. Cir. 2012); *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1333-1334 (Fed. Cir. 2012); *Cyber-*

IV. The Claims In This Case Are Not Patent Eligible.

This is a straightforward case for applying Section 101. The claims here do not assert a process accomplishing something unique to a digital environment. Instead, they merely purport to take an abstract business method—the abstract idea of intermediated settlement—and “apply it” on a computer.

The claims do not contain limits that would transform the abstract idea into a practical, patent-eligible application. Rejecting these claims, which are effectively “the *Bilski* claims on a computer,” says nothing at all about the patentability of the real software innovations that power the Information Age.

A. Intermediated Settlement Is An Abstract Idea.

Petitioner’s claims implement the concept of intermediated settlement. This is a “fundamental * * * practice,” understood for centuries, if not millennia, and no different than the practice of risk hedging. *Bilski*, 130 S. Ct. at 3231. It is thus an “abstract idea.” *Ibid.*

This is not a controversial proposition: nine of the ten members of the *en banc* agreed so expressly. The five-judge plurality, authored by Judge Lourie, noted that the patent claims “draw on the abstract idea of reducing settlement risk by effecting trades through a third-party intermediary.” Pet. App. 28a. Judges Rader and Moore (*id.* at 82a) as well as Judges Linn and O’Malley (*id.* at 124a) agreed.

Source Corp. v. Retail Decisions, Inc., 654 F.3d 1366, 1373 (Fed. Cir. 2011).

Petitioner in this Court substantially revises its argument that intermediated settlement is not an abstract idea. Seizing the statement in *Bilski* that risk hedging could be “reduced to a mathematical formula,” petitioner argues that an “abstract idea” is *only* “[a] fundamental truth in the sense of a mathematical formula [that] reveals a relationship that has always existed.” Pet. Br. 22 (quotation omitted). Petitioner suggests that the court below has improperly conflated “disembodied concept[s]” with “abstract ideas.” *Id.* at 27. Petitioner ultimately contends that because its claims do not recite “mathematical formulas and the like,” they do not state an abstract idea at all. See *id.* at 21-29, 44-46.

Petitioner’s argument is wrong for at least two reasons: this Court’s decisions require rejection of the contention that abstract ideas are limited to “fundamental truths,” and many abstract ideas cannot be reduced to mathematical formulae.

First, Bilski held that “the basic concept of hedging” constituted an “abstract idea” because it was a “fundamental economic practice[s] long prevalent in our system of commerce and taught in any introductory finance class.” 130 S. Ct. at 3231 (quotation omitted). But risk hedging is not a naturally occurring phenomena that humans one day observed; it is a concept that, at some point millennia ago, humans devised. *Ibid. Bilski* thus establishes that the category of unpatentable abstract ideas is not limited to “fundamental truths” that have “always existed.” Under petitioner’s theory, by contrast, *Bilski* was wrongly decided.

Petitioner’s theory is also foreclosed by *O’Reilly v. Morse*, 56 U.S. 62, 113 (1853). In famously barring Samuel Morse from patenting the use of “electro-

magnetism, however developed” for sending signals “at any distances,” the Court explained that this was merely an effort to patent a concept. *Id.* at 112-113. The Court drew a hypothetical involving steam-powered transportation: “No one, we suppose will maintain that Fulton could have taken out a patent for his invention of propelling vessels by steam, describing the process and machinery he used, and claimed under it the exclusive right to use the motive power of steam, however developed, for the purpose of propelling vessels.” *Id.* at 113.

Morse thus made clear that one cannot patent the mere concept of using a steam engine to power transportation. But such a concept, just like risk hedging, is hardly a “fundamental truth” that reveals some “relationship that has always existed.” Pet. Br. 22 (quotation omitted). It is an abstract idea created by humans, but nonetheless ineligible for patent protection.⁷

Second, some unpatentable abstract ideas are not reducible to a mathematical formula. There is no shortage of abstract concepts that have existed for millennia and are ineligible for patent protection, but

⁷ Petitioner argues it would make “little sense” if an “abstract idea” implicitly exempt from Section 101 could be in the form of a “pure mental conception” because such a concept, “standing alone, does not fall within any of the four statutory categories” of Section 101. Pet. Br. 27-28. But, as the Court held more than a century ago, a “process” may be eligible for patent protection even if it is “a conception of the mind, seen only by its effects when being executed or performed.” *Tilghman*, 102 U.S. at 728. Moreover, any abstract idea could be written as a process claim to perform that abstract idea, with nothing more; Section 101 bars such a patent.

that are not reducible to some mathematical formula.

For example, the concept of “money” is an abstract idea; one could not patent the idea of using coins, banknotes, or the like as a medium of exchange. But the idea of “money” is not reducible to a mathematical formula. “Advertising” is an abstract idea. The idea of “credit” is abstract; so too is “trusteeship.” One could not patent these concepts, nor countless others, as such. (Of course, novel and limited *applications* will qualify as patent-eligible.)

Petitioner contends that the lower court erred by stripping the concept “from any real-world application,” because its claim provided the application. Pet Br. 46-47 (quotation and emphasis omitted). But this point has no bearing on whether intermediated settlement is an abstract concept. Because it is, the Court must next ask, “What else is there in the claims before us?” *Mayo*, 132 S. Ct. at 1297. The answer here is clear: nothing.

B. The Patents Contain No Other Meaningful Limitations.

Because petitioner’s claim involves computer-implementation of an abstract idea, the next step of the Section 101 analysis is to determine whether the claim “in practice amounts to significantly more than a patent upon” the abstract idea. *Mayo*, 132 S. Ct. at 1294. Petitioner’s claims here do nothing more than claim intermediated settlement implemented in any manner on a computer.⁸ This is not enough to transform the abstract idea into a patent-eligible claim.

⁸ Petitioner’s method claims do not even require, on their face, the use of a computer. See, *e.g.*, JA 383-384. Although

In fact, petitioner does not even attempt to disguise that this is all its claims do. It says the relevant claim limitation is “requiring computer implementation,” and that the innovation it asserts is that the “computer *is itself the intermediary*.” Pet. Br. 48. But taking an abstract idea, saying “apply it on a computer,” and adding nothing more is not enough to overcome the implicit exceptions to Section 101. See pages 27-29, *supra*.

Petitioner contends that the claims “include steps or elements other than those necessary to any use of the idea.” Pet. Br. 49. But all these steps do is recite how one can implement intermediated settlement. This is no different than *Mayo*, where the patentee provided a series of steps that, because “[a]nyone who wants to make use of these laws must” perform them, the asserted “combination amounts to nothing significantly more than an instruction to doctors to apply the applicable laws when treating their patients.” 132 S. Ct. at 1298.

Seven judges below endorsed this conclusion in analyzing petitioner’s process claim. The plurality succinctly summarized the purported limitations and correctly found that “[n]one of those limitations adds anything of substance to the claim.” Pet. App. 29a. Judges Rader and Moore likewise concluded that “each step individually recites merely a general step inherent within the concept of an escrow.” *Id.* at 83a.

the parties below agreed that there is an implicit computer limitation arising from the specification (Pet. App. 27a-28a), the fact that a computer is not even disclosed in several of the asserted claims underscores that this is just an abstract business method dressed up as software.

C. The Form Of The Claims Does Not Alter The Result.

Finally, the form of the patent claim has no bearing. As is typically the case of software-related innovations, petitioner holds three different kinds of claims: method claims (which direct a particular process to be accomplished via software, often via an algorithm), system claims (which disclose certain kinds of computer hardware that would be used to perform a process), and media claims (which are computer data storage devices programmed with software code, sometimes referred to as “*Beauregard* claims”). Pet. Br. 6-10.

This Court has repeatedly made clear that application of Section 101 cannot “depend simply on the draftsman’s art.” *Mayo*, 132 S. Ct. at 1294; *Flook*, 437 U.S. at 593. Because method claims, system claims, and media claims are functionally equivalent—“it is often a straightforward exercise to translate a method claim into system form, and vice versa” (Pet. App. 36a)—there is no basis to consider these claims separately.

Eight members of the court below concluded that the form of the claim has no bearing on its eligibility for patent protection under Section 101. Pet. App. 3a n.1. The plurality noted that the media claims, for example, “are merely method claims in the guise of a device.” *Id.* at 34a. Judges Linn and O’Malley (*id.* at 123a) as well as Judge Newman (*id.* at 102a) agreed.

Petitioner hardly disputes this point. It says nothing at all about its media claim. As to its system claims, petitioner argues that, even assuming intermediated settlement is an abstract idea, the system claims are patent-eligible because they “recite specif-

ic hardware, configured to perform the specific computerized functions.” Pet. Br. 53. But the computer hardware devices it discloses—which includes “a data storage unit” and a “computer” (Pet. App. 35a)—are in no way “specific.” Those devices are so general (indeed they describe any off-the-shelf personal computers) that the effect would be to preempt *every* computer implementation of the idea.

Under petitioner’s view, therefore, a system claim that provides a process to calculate $E=mc^2$, coupled with disclosure of a device called a “computer,” would be enough to patent the concept. But this would amount to an astounding circumvention of Section 101, permitting a party to patent *any* law of nature, physical phenomena, or abstract idea via computer implementation, so long as rote, meaningless formalities—*i.e.*, saying one must use a “computer”—are attached.

CONCLUSION

The judgment of the court of appeals should be affirmed.

Respectfully submitted.

ANDREW J. PINCUS
Counsel of Record
PAUL W. HUGHES
Mayer Brown LLP
1999 K Street, NW
Washington, DC 20006
(202) 263-3000
apincus@mayerbrown.com

Counsel for Amicus Curiae

FEBRUARY 2014