No. 08-964

IN THE
Supreme Court of the United States

BERNARD L. BILSKI AND RAND A. WARSAW,
Petitioners,
v.
DAVID J. KAPPOS, UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND DIRECTOR, PATENT AND TRADEMARK OFFICE,
Respondent.

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

BRIEF OF THE SOFTWARE FREEDOM LAW CENTER AS *AMICUS CURIAE* IN SUPPORT OF RESPONDENT

EBEN MOGLEN
Counsel of record
DANIEL RAVICHER
MICHAEL A. SPIEGEL
MISHI CHOUDHARY
AARON WILLIAMSON
Software Freedom Law Center
1995 Broadway, 17th Floor
New York, New York 10023
212-461-1900

October 1, 2009

QUESTION PRESENTED

Was the Court of Appeals correct in holding that software standing alone is not patentable subject matter?

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

Constitutions, Statutes, and Rules

Other Authorities

INTEREST OF THE AMICUS CURIAE

Much of the world's most important and most commercially significant software is distributed under copyright licensing terms that give recipients freedom to copy, modify and redistribute the software ("free software").[1] One could not send or receive e-mail, surf the World Wide Web, perform a Google search or take advantage of many of the other benefits offered by the Internet without free software. Indeed, this brief was written entirely with free software word processors, namely OpenOffice, GNU Emacs and LaTeX, each of which are not just competitive with or superior to non-free software programs, but which also provide their users with the freedom to improve the program to fit their needs and reflect their desires.

The Software Freedom Law Center ("SFLC") is a not-for-profit legal services organization that provides legal representation and other law-related services to protect and advance free software. SFLC provides pro bono legal services to non-profit free software developers and also helps the general public better understand the legal aspects of free software. SFLC has an interest in this matter because the decision of this Court will have a significant effect on the rights of the free software developers and users SFLC represents. More specifically, SFLC has an interest in ensuring

[1] Pursuant to Sup. Ct. R. 37.6, *amicus* notes that no counsel for a party authored this brief in whole or in part, and no counsel or party made a monetary contribution intended to fund the preparation or submission of this brief. No person other than *amicus curiae* and its counsel made a monetary contribution to its preparation or submission. Petitioners and Respondents have consented to the filing of this brief through blanket consent letters filed with the Clerk's Office.

that limits are maintained on the reach of patent law so that Free and Open Source software development is not unreasonably and unnecessarily impeded.

SUMMARY OF ARGUMENT

In *Microsoft v. AT&T*, this Court recognized that "[a]bstract software code [uninstalled in a machine] is an idea without physical embodiment." 550 U.S. 437, 449 (2007). The court below correctly decided that, on the basis of this Court's prior holdings, such abstract ideas without physical embodiment cannot be the subject of a statutory patent monopoly.

Software standing alone cannot be patentable subject matter under 35 U.S.C. § 101, because, as this Court has repeatedly recognized in an unbroken series of cases extending over more than 130 years, "An idea of itself is not patentable." *Rubber-Tip Pencil v. Howard*, 87 U.S. (20 Wall.) 498, 507 (1874). The court below correctly held, therefore, that the Patent Act should be interpreted so that computer software can contribute to the claims of a patent only to the extent that this software is combined with a special-purpose machine or is used in a "process" that transforms matter, like the rubber undergoing vulcanization in *Diamond v. Diehr*, 450 U.S. 175 (1981). The Federal Circuit's conclusion on this point is the only outcome consistent with the prior teachings of this Court. It also represents the outcome best calculated to produce maximum innovation in the production of computer software under contemporary technical conditions. Any other holding would not only breach the line previously established by the decisions of this Court between patentable art and processes on the one hand and abstract ideas or facts of nature on the other—in doing so it would also raise grave constitutional issues under the analysis set forth by this Court in *Eldred v. Ashcroft*, 537 U.S. 186 (2003).

I. Software Is Algorithms For Computers In Human Readable Terms, And Is Therefore Not Patentable, As This Court Has Repeatedly Held

The Court of Appeals for the Federal Circuit held below that the presence of a particular machine or apparatus or the transformation of a particular article into a different state or thing turns the abstract ideas expressed by a computer program code into a "process" within the meaning of the Act. This conclusion correctly follows from this Court's prior holdings.

This Court has repeatedly held that subject matter which would have the practical effect of preempting laws of nature, abstract ideas or mathematical algorithms is ineligible for patent protection. Further, in *Microsoft v. AT&T* you also stated that abstract software code is an idea without physical embodiment and is merely information—a detailed set of instructions. A computer program, no matter what its function, is nothing more or less than the representation of an algorithm. It is not conceptually different from a list of steps written down with pencil and paper for execution by a human being. In no uncertain terms, this Court in *Gottschalk v. Benson*, 409 U.S. 63 (1972), held that software, which contains and upon command executes algorithms that solve mathematical problems through the use of a computer, was not patentable under § 101.

Thus, software standing alone, without the presence of a special purpose machine or the act of transforming a particular article into a different state or thing is merely information, a representation of an algorithm or algorithms, and not a "process" within the meaning of this Act.

Confronted with the rise of new technologies, this Court has addressed the issue of patentable subject matter several times. *Benson*, 409 U.S. 63; *Parker v. Flook*, 437 U.S. 584 (1978); *Diamond v. Chakrabarty*, 447 U.S. 303 (1980); *Diamond v. Diehr*, 450 U.S. 175 (1981).

Since before the Civil War, this Court has consistently made it clear that subject matter which would have the practical effect of preempting laws of nature, abstract ideas or mathematical algorithms is ineligible for patent protection. *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 113 (1854); *Benson*, 409 U.S. at 71. These cases effectively establish a clear boundary of ineligibility for patent protection to safeguard the fundamental constitutional requirement that laws of nature, abstract ideas, mathematical propositions and algorithms be left unrestrained by patents.

This Court stated in *Flook* that to be eligible for patent protection, "[a] process itself, not merely the mathematical algorithm, must be new and useful." 437 U.S. at 591, observing also that it is "incorrect [to] assume[] that if a process application implements a principle in some specific fashion, it automatically falls within the patentable subject matter of § 101... ." 437 U.S. at 593. Such an assumption, the Court said, is based on an impermissibly narrow interpretation of prior cases, including specifically *Benson,* and is "untenable" because "[i]t would make the determination of patentable subject matter depend simply on the draftsman's art and would ill serve the principles underlying the prohibition against patents for 'ideas' or phenomena of nature." *Id.*

In alignment with *Benson* and *Flook*, this Court's decision in *Diamond v. Diehr* held that structures or processes must, when considered as a whole, perform functions intended to be covered by patent law in order to be eligible for patent protection. 450 U.S. at 192. *Diehr* followed and upheld the core holdings of both *Benson* and *Flook*. *Id.* at 191–193 ("[o]ur reasoning in *Flook* is in no way inconsistent with our reasoning here").

Benson, *Flook*, *Diehr*, and the other decisions of this Court regarding patentable subject matter consistently established that the inquiry into whether subject matter is eligible for patenting is one of substance, not form. This Court requires that one look not simply at the language of the patent claim to see if it recites a structure of multiple steps or components, but also at the practical effect of the claim to see if it in fact covers—or otherwise would restrict the public's access to—a principle, law of nature, abstract idea, mathematical formula, mental process, algorithm or other abstract intellectual concept.

This substantive standard ensures that skilled patent draftsmanship is not capable of overcoming one of the core doctrines of patent law recognized by this Court for more than 150 years that "[a] principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right." *Le Roy v. Tatham*, 55 U.S. (14 How.) 156, 175 (1853); *Funk Bros.*, 333 U.S. 127, 130 (1948); *Benson*, 409 U.S. at 67 ("[p]henomena 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"). Despite the arguments made by Petitioners and their *amici* to the contrary, this Court has clearly and unambiguously limited the scope of patentable subject matter to exclude software from its ambit in the most definite terms.

In *Benson*, this Court held invalid a software patent directed to a specific application of a generic formulation because "the mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself." 409 U.S. at 71–72.

The holding of *Benson* is properly applicable to all software, because a computer program, no matter what its function, is neither more nor less than the representation of one or more algorithms. Further, just as claiming fifty—or even a thousand—laws of nature is no more patentable than claiming a single law of nature, no form of software, regardless of how many algorithms or formulas it comprises, is patentable: It will always be merely and solely made up of algorithms for computers to execute, written down in a form comprehensible to human beings for them to study and expand upon.

This Court's decision in *Diehr* is not to the contrary. It followed the teaching of *Benson*, and determined that the invention there claimed was not substantially the software, but rather the totality of an "industrial process for the molding of rubber products," which is undeniably included within the realm of patentable subject matter. 450 U.S. at 191–93. Had the applicant sought to claim the software used in that process by itself, however, this Court would have most assuredly found it to be unpatentable subject matter just as it had in *Benson*.

In *Diehr*, this Court observed:

> When a claim recites a mathematical formula (or scientific principle or phenomenon of nature), an inquiry must be made into whether the claim is seeking patent protection for that formula in the abstract. A mathematical formula as such is not accorded the protection of patent laws, and this principle cannot be circumvented by attempting to limit the use of the formula to a particular technological environment.

450 U.S. at 191 (internal citation omitted). This result—which makes software describing a portion of the solution to a practical problem unpatentable on its own, outside the real-world context of the problem and its solution—is not only in accord with the rest of this Court's patent jurisprudence, it is also the best way to protect innovation in software and the only way that fully comports with both Article I, Section 8 and the First Amendment.

II. Excluding Software From Patentable Subject Matter Maximizes Innovation In Software

If mathematics were patentable, there would be less mathematical innovation. Only those who were rich enough to pay royalties, or who benefited from subsidization by government, or who were willing to sign over the value of their ideas to someone richer and more powerful than themselves, would be permitted access to the world of abstract mathematical ideas. Theorems build upon theorems, and so the contributions of those who could not pay rent—and all the further improvements based upon those contributions—would be lost.

For more than a quarter century, beginning with a few stalwart thinkers and exponentially increasing in size and influence, a movement to build computer software by sharing—treating software programming languages like mathematical notation, for the expression of abstract ideas to be studied, improved, and shared again—has revolutionized the production of software around the world. The "free software movement," believes, like this Court, that computer software expresses abstract ideas, and therefore concludes that the ideas themselves will grow best if left most free to be learned and improved by all. It has attracted hundreds of thousands, soon millions of programmers around the world to the making of new and innovative software through the social process that for centuries has been the heart of Western science: "share and share alike."

Free software, often referred to by commercial entities as "open source software," to prevent confusion between the social freedom of its making and the price at which they sell copies, has become the single most influential body of software around the world. Free software operating systems now power everything from cellphones to home appliances to supercomputer clusters. There is no major or minor computer hardware architecture, no class of consumer electronics, no form of network hardware connecting humanity's telephone calls, video streams, or anything else transpiring in the network of networks we call "the Internet" that doesn't include free software. The most important innovations in human society during the last decade, the World Wide Web and the Wikipedia, were based on and are now dominated by free software and the idea of free knowledge sharing it represents.[2]

This explosion of technical innovation has occurred for two primary reasons. First, the principle rule of free software, the required sharing of computer program source code, has allowed young people around the world to learn and apply their skills by studying and improving real software doing real jobs in their own and others' daily lives. This process has enabled the incremental improvement of the art by everyone, rather than by the necessarily small number of programmers working for any one firm with proprietary control of source code. Second, by creating a "protected commons" for the free exchange of ideas embodied in program source code, the free software copyright licensing structure has enabled cooperative interactions among competing firms: each firm has been assured of permanent continuing access to the improvements in program code made by all other firms, which were required to make the source code of those im-

[2]The brief submitted by *amici* Lee A. Hollar and IEEE is therefore entirely wrong in claiming that the free software movement's hostility to software patenting results from a similarity between "underdeveloped economies" and free software programmers. Brief of Lee A. Hollar and IEEE-USA at 21 n.31. Far from being involved in "cloning" the software of others, free software or "open source" programmers have been responsible for the most important technical innovations in the field over the last twenty years. The objection of the free software community to software patents stems not from their desire to misappropriate the work of others, but rather from the belief that the free exchange of information is a cornerstone of free societies.

provements freely available to all users. Thus firms were able to increase their levels of investment in cooperative production, and were able to exchange ideas with academic researchers, secure in the knowledge that those investments would not be appropriated by others claiming exclusive rights.

The principle that innovation is made possible by the free exchange of ideas is not recent, and is not limited to software. Indeed, our constitutional system of free expression since Thomas Jefferson is based on the recognition that control of ideas by power has never produced more ideas than their free and unrestricted circulation. The history of western science since the 17th century is one long testament to this truth, and it is that very history which gave rise to the patent system, whose exclusion of "abstract ideas," "facts of nature," and "algorithms" is as much as recognition of the principle as is the basic constitutional policy of offering temporary legal benefits in return for prompt and complete disclosure of technological discoveries to the public.

This Court has recognized the growth and innovation in the software industry in the absence of patent protection. In *Benson*, this Court noted that "the creation of programs has undergone substantial and satisfactory growth in the absence of patent protection and that copyright protection for programs is presently available." 409 U.S. at 72 (quoting "To Promote the Progress of . . . Useful Arts," Report of the President's Commission on the Patent System (1966)). A few years later, in *Diehr*, the Court observed that "[n]otwithstanding fervent argument that patent protection is essential for the growth of the software industry, commentators have noted that 'this industry is growing by leaps and bounds without it.'" 450

U.S. at 217 (internal citations omitted). Recent empirical data also suggests that software patents actually stifle innovation instead of promoting it. See J. Bessen and M. Meurer, *Do Patents Perform Like Property?*, Academy of Management Perspectives, pp. 8–20 (Aug. 2008) ("The direct comparison of estimated net incentives suggests that for public firms in most industries today, patents may actually discourage investment in innovation.")

III. The First Amendment Prohibits Construing The Patent Act To Permit The Patenting Of Abstract Ideas

This Court held in *Eldred v. Ashcroft*, 537 U.S. 186 (2003), that the First Amendment precludes the extension of statutory monopolies to abstract ideas. *Eldred*, 537 U.S. at 219. As you there said, the near-simultaneous adoption of the Patent and Copyright Clause and the First Amendment indicates that these laws are fundamentally compatible. This compatibility, however, depends on a construction of the patent and copyright laws that preserves First Amendment principles, including the freedom to communicate any "idea, theory, and fact." *Id.*

Eldred identified two mechanisms in copyright law that are necessary to accommodate this principle. First, the idea/expression dichotomy limits copyright's monopoly to an author's expression, leaving ideas "instantly available for public exploitation." *Id.* Second, the fair use doctrine allows the public to use even copyrighted expression for some purposes, "such as criticism, comment, news reporting, teaching..., scholarship, or research." *Id.* at 220.

Patent statutes, which depend on the same constitutional grant of authority as copyright statutes are similarly limited by the First Amendment. *See Eldred*, 537 U.S. at 201 ("Because the Clause empowering Congress to confer copyrights also authorizes patents, congressional practice with respect to patents informs our inquiry."). The presence of an unwavering exemption for abstract ideas reconciles patent law with the First Amendment in a fashion similar to the idea/expression dichotomy's crucial role in reconciling copyright and freedom of speech. The presence of a limiting principle is even more necessary with respect to patent law than with respect to copyright, because, as you observed in *Eldred*, "the grant of a patent... prevent[s] *full use* by others of the inventor's knowledge." 537 U.S. at 217 (emphasis added) (internal citation omitted). Patents can and do limit the application of knowledge to produce a new machine or to transform an article into a different state or thing, but they cannot constitutionally limit the communication of knowledge or ideas. *Eldred* teaches that, without this limitation, determining the scope of patent eligibility in each individual case would raise First Amendment questions of great difficulty. Patent law also recognizes no analogue to fair use, previously described by this Court as the second bulkwark of constitutional harmony between copyright and free expression. 537 U.S. at 219-220. The absence of any provision for fair use substantially increases the constitutional difficulty when patents are sought and granted for expressions of abstract ideas.

Therefore, patents on software standing alone, which could be infringed by the dissemination of computer program source code as well as by the execution of such code by a special-purpose tool or in the course of an industrial process with a material result, disturb the "definitional balance" between the First Amendment and the Patent Act. *Eldred*, 537 U.S. at 219. In its unprocessed source code form, software is merely the expression of abstract ideas in human language—a description of a sequence of steps that will produce a particular result (i.e. an "algorithm"). The source code of a program which performs the steps described in a software patent is distinguishable from the literal patent only in that it expresses the same steps in a different language. Therefore, since anyone may copy or publish the actual patent without infringing, it must also be permissible to communicate its claims in source code form.

The sharing of source code is also essential to "scholarship and comment," two categories of speech recognized in *Eldred* and *Harper & Row, Publishers v. Nation Enters.*, 471 U.S. 539, 560 (1985), as particular First Amendment concerns. Computer science textbooks, for example, rely heavily on source code and pseudo-code to communicate concepts and describe useful algorithms. *See, e.g.,* Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language* (Prentice Hall 1978). Likewise, computer science students are often required to express their answers to test questions in a real or hypothetical programming language. And without the use of source code, it is difficult for developers to comment on whether an idea can be implemented, to comment on an algorithm's performance, or to suggest improvements.

Moreover, First Amendment concerns cannot be avoided by exempting source code and limiting patent protection to software which has been converted to executable form, or which has actually been executed. Recent rulings, such as the Federal Circuit's decision in *Metabolite Labs, Inc. v. Laborator Corp. of Am. Holdings*, 370 F.3d 1354 (Fed. Cir. 2004), *cert. dismissed*, 548 U.S. 124 (2006), demonstrate that secondary liability would still restrict the communication of ideas expressed in source code. In *Metabolite*, a medical testing laboratory was successfully sued for inducement to patent infringement for telling doctors what steps to take to correlate the presence of amino acids in body fluids with a vitamin deficiency. 370 F.3d at 1358. Once communicating the content of a patent to doctors is held inducement to patent infringement, constitutionally-protected communication among programmers will inevitably be chilled.

This result would be a restriction on the expression of abstract ideas. The court below correctly interpreted the Act to avoid this dubious outcome, by requiring more than an abstract expression of an idea: either a machine specially adapted to the implementation of that idea, or a transformation of the physical environment mediated or controlled by software. Any other construction of the statute would raise serious constitutional questions.

CONCLUSION

For the foregoing reasons, the decision below should be affirmed.

Respectfully submitted.

EBEN MOGLEN
Counsel of record
DANIEL RAVICHER
MICHAEL A. SPIEGEL
MISHI CHOUDHARY
AARON WILLIAMSON
Software Freedom Law Center
1995 Broadway, 17^{th} Floor
New York, New York 10023
212-461-1900

October 1, 2009