

No. 10-6

IN THE
Supreme Court of the United States

GLOBAL-TECH APPLIANCES, INC., *et al.*,
Petitioners,

v.

SEB S.A.,
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 PETITIONERS

EBEN MOGLEN

Counsel of record

DANIEL RAVICHER

AARON WILLIAMSON

Software Freedom Law Center

1995 Broadway, 17th Floor

New York, New York 10023

moglen@softwarefreedom.org

212-461-1900

December 6, 2010

QUESTION PRESENTED

Whether the Federal Circuit's "deliberate indifference" standard should supplant this Court's standard for the state of mind element of an inducement claim?

TABLE OF CONTENTS

TABLE OF AUTHORITIES	ii
INTEREST OF THE AMICUS CURIAE	1
SUMMARY OF ARGUMENT	2
ARGUMENT	3
I. Relaxing the the <i>mens rea</i> for Inducement Threatens Free Software Development and Therefore the Entire Software Economy	3
II. Secondary Liability is the Sole Source of Potential Patent Liability for Source Code—Which Embodies Abstract Ideas	6
III. Federal Circuit’s Standard Burdens Free and Open Source Software Developers, Chilling Progress and Innovation	8
CONCLUSION	12

TABLE OF AUTHORITIES

Cases

<i>Bilski v. Kappos</i> , 130 S. Ct. 3218 (2010)	7, 10
<i>Diamond v. Diehr</i> , 450 U.S. 175 (1981)	7
<i>Gottschalk v. Benson</i> , 409 U.S. 63 (1972)	6, 7
<i>MGM Studios, Inc. v. Grokster, Ltd.</i> , 545 U.S. 913 (2005)	8, 9
<i>Microsoft v. AT&T</i> , 550 U.S. 437 (2007)	6-7
<i>O'Reilly v. Morse</i> , 56 U.S. (15 How.) 62 (1854)	6
<i>Parker v. Flook</i> , 437 U.S. 584 (1978)	6
<i>SEB SA v. Montgomery Ward Co. Inc.</i> , 549 F.3d 1360 (Fed. Cir. 2010)	8-10

Constitutions, Statutes, and Rules

26 U.S.C. § 501(c)(3)	4
35 U.S.C. § 101	6, 7
35 U.S.C. § 271(b)	8

Sup. Ct. R. 37.6 1

Other Authorities

James Bessen & Michael J. Meurer, *Patent Failure:
How Judges, Bureaucrats, and Lawyers
Put Innovators at Risk* (2008) 10

INTEREST OF THE AMICUS CURIAE

Much of the world's most important and 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.org, GNU Emacs, and L^AT_EX, each of which are not just competitive with or superior to non-free software programs, but 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

¹Pursuant 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 development of free software is not unreasonably and unnecessarily impeded.

SUMMARY OF ARGUMENT

The Federal Circuit's new standard for induced infringement could expose free software developers to liability from which this Court's standard has traditionally shielded them. This risk would discourage developers from developing free software, impeding the growth of a valuable public good: software that anyone can freely use, copy, modify, and distribute.

Under current law, developers are not subject to direct liability for patent infringement when they produce and distribute free software source code, because computer program source code is merely the expression of abstract ideas, and therefore not itself patentable. These developers' only potential source of liability, then, has been for secondary liability, including for inducing infringement. But the Federal Circuit's standard of culpable knowledge sets the bar to an inducement claim so low as to allow patentees to circumvent the statutory limit on patentability of abstract ideas. Merely by writing software, a developer might become vulnerable to an inducement claim; the patentee would only need to allege some vague awareness of neighboring patents.

In addition to subverting the current limits on patentability, the Federal Circuit's "deliberate indifference" standard replaces the certainty of the previous standard with unknowable liability. A *mens rea* standard requiring "actual knowledge" allows devel-

opers to easily understand their risk of inducing infringement: it is circumscribed by the patents of which they are aware. Most free software developers receive no compensation for their contribution. They cannot afford to retain counsel to evaluate the patent risk of each new improvement they make, or to review all four thousand patents issued every week to be sure that none threaten their work. An indefinite “deliberate indifference” standard might compel them—and indeed all software developers—to do just that.

Free software is particularly vulnerable to this new, uncertain risk. Many developers, faced with the unascertainable (and potentially ruinous) liability imposed on them by the Federal Circuit’s standard, may simply stop producing free software for fear of being harassed by patent trolls who sue them for inducing infringement by not thoroughly vetting all patents before releasing new code. And because free software is fundamental to the construction of the Internet, modern consumer electronics, and the information technology infrastructure of nearly every large business, that loss will be devastating to the entire software economy.

ARGUMENT

I. Relaxing the the *mens rea* for Inducement Threatens Free Software Development and Therefore the Entire Software Economy

Free software is produced collaboratively and licensed freely on terms that grant others the freedom to use, copy, modify, and redistribute the software for no cost. This “free software movement” has attracted

countless programmers from 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 software² has become arguably 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 telephone calls, video streams, or anything else transpiring in the network of networks we call “the Internet” that doesn’t rely on free software.

Two of 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. Much of this free software development has been undertaken by 26 U.S.C. 501(c)(3) recognized non-profit organizations, developing software as a public good.

This explosion of technical innovation has been made possible for two primary reasons, both jeopardized by the Federal Circuit’s deviation from settled law in the case below. First, the principle rule of free software—the sharing of computer program source code—has allowed young people 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

²Free software is sometimes referred to as “open source software.”

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 copyright licensing innovation known as copyleft—which requires those who take advantage of the grant to use, copy, and redistribute the software to perpetuate these freedoms—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 improvements freely available to all users. Thus firms are able to increase their levels of investment in cooperative production and to exchange ideas with academic researchers, secure in the knowledge that those investments would not be appropriated by others claiming exclusive rights.

Free software continues to propel innovation today, allowing software developers to build atop existing source code instead of starting from scratch. This gives users real, immediate power: with access to source code and the right to modify it, developers and users can make the software more useful, remake it into something wholly new, or even turn it into a competing product. For example, there are dozens of successful operating systems built on top of the Linux kernel, such as Red Hat and Novell server systems, Nokia and Google mobile phone operating systems, as well as non-commercial desktop operating systems such as Debian and Ubuntu. Because these rights that free software developers share so freely are ones that competitive entities regard as valuable property, many firms do not offer them to end users at any price.

The Federal Circuit’s opinion would likely expand

the patent liability risk to free software. This expansion could chill free software development, discouraging developers, users and firms from producing software that contains these fundamental freedoms, threatening the social and societal benefits that free software provides.

II. Secondary Liability is the Sole Source of Potential Patent Liability for Source Code—Which Embodies Abstract Ideas

Not only does the Federal Circuit’s opinion below expand liability generally, it also allows patent holders to reach beyond the limitations this court has consistently imposed on patent eligibility under 35 U.S.C. § 101. Merely by alleging that software developers were deliberately indifferent to their patents, patentees may now use inducement to circumvent the limits long held by this Court that “abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work.” *Gottschalk v. Benson*, 409 U.S. 63, 67 (1972). To allow the decision below to stand would undermine “the principles underlying the prohibition against patents for ‘ideas’ or phenomena of nature.” *Parker v. Flook*, 437 U.S. 584, 593 (1978).

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. See *O’Reilly v. Morse*, 56 U.S. (15 How.) 62, 113 (1854); *Benson*, 409 U.S. at 71. Further, in *Microsoft v. AT&T*, this Court stated that abstract software code is an idea without physical em-

bodiment and is merely information—a detailed set of instructions. 550 U.S. 437, 449–450 (2007). A computer program in source code form, 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 *Benson* held that software, which contains and upon command executes algorithms that solve mathematical problems through the use of a computer, is not patent-eligible under 35 U.S.C. § 101. Most recently, *Bilski v. Kappos* reaffirmed these cases by holding that both conceptual and formulaic representations of hedging algorithms were not patentable. 130 S. Ct. 3218, 3231 (2010).

However, that computer program source code standing alone cannot infringe patents does not mean that software developers are insulated from patent liability. Indeed, as this Court taught in *Diamond v. Diehr*, structures or processes may be eligible for patent protection, when considered as a whole, if they perform functions intended to be covered by patent law. Therefore software—embodying abstract ideas—may infringe a patent only when it is rendered executable, and is then used in conjunction with other structures or processes. This result means that producers and distributors of free software, in the absence of personal use, cannot directly infringe patents by merely writing and distributing source code, and so can be subject only to secondary liability.

Until the recent decision by the Federal Circuit, software source code that, for example, could be “compiled” into an executable program that merely read inputs and computed the equation at issue in *Diehr* could not induce infringement of the relevant patent

in absence of industrial application unless the producer possessed knowledge of the patent and was shown to have used “purposeful, culpable expression and conduct” to encourage use of the software to mold rubber. *See MGM Studios, Inc. v. Grokster, Ltd.*, 545 U.S. 913, 937 (2005). However, the Federal Circuit decision below casts aside this clear standard—opening software producers to unforeseen liability and chilling their incentive to continue producing high quality software free for all to use, modify, and share.

Subjecting software developers to the risk of secondary patent infringement liability renders meaningless their existing protection from risk of direct infringement, because the remedies available to a patentee for inducement are the same. *See* 35 U.S.C. § 271(b). Thus, a patentee can threaten developers with secondary liability even in the absence of a claim of direct patent infringement for the making or selling of the software. This result renders the Court’s limitations on patentable subject matter pragmatically irrelevant to working developers, and the Court’s defense of pure source code from patenting impotent. The decision of the court below would allow patentees to chill conduct through allegations of inducement of infringement that they could not threaten directly.

III. Federal Circuit’s Standard Burdens Free and Open Source Software Developers, Chilling Progress and Innovation

The Federal Circuit’s holding below decreases the standard of *mens rea* for inducing patent infringement from knowing, purposeful conduct to “deliberate indifference of a known risk.” *SEB SA v. Montgomery*

Ward & Co. Inc., 594 F.3d 1360, 1377 (Fed. Cir. 2010). This standard, easier for plaintiffs to meet,³ potentially exposes each tinkerer, student, or volunteer programmer—as well as every corporation employing free software engineers—to unforeseeable and unmanageable amounts of liability.

Indeed, free software development is particularly chilled by this standard. Unlike typical production schemes, free software is usually produced by a loose coalition of volunteers, who are otherwise unassociated. For inducement claims, the *mens rea* requirement mitigated this liability, by holding alleged inducers responsible only for purposeful conduct. This mitigating factor, should the Federal Circuit’s test stand, would be severely weakened, potentially exposing each developer to uncertain liability, and chilling participation. Because contributors do not have the resources to conduct an independent patent search on parts of the code with which they may not be familiar, they cannot evaluate their risk in contributing code to a free software project. Many developers of free software derive no compensation for development and distribution of the software, and might stop participating rather than risk liability. This Court should reject the Federal Circuit’s test and reaffirm *Grokster’s* test, which clearly limits liability to purposeful conduct to induce others.

It has already been empirically demonstrated that applying patents to software has an economically negative effect and results in a net drag on innova-

³Indeed, the case below held that the mere fact of an officer of the defendant’s firm being well versed in patent law, taken together with other factors, was supported a finding of deliberate indifference and thus inducement of infringement liability. *Montgomery Ward*, 594 F.3d at 1377.

tion in the field. See James Bessen & Michael J. Meurer *Patent Failure: How Judges, Bureaucrats, and Lawyers Put Innovators at Risk* (2008). The same social waste results from expanding secondary liability to reach software developers in the absence of active steps to induce infringement.

Adding to the net economic injury of subjecting software source code to potential liability through the back door of inducement, the Federal Circuit creates economic harm through uncertainty by failing to notify the public of what standard of knowledge results in liability, noting that it did not “purport to establish the outer boundaries of knowledge needed for inducement.” *SEB SA v. Montgomery Ward & Co. Inc.*, 594 F.3d 1360 1378 (Fed. Cir. 2010). This uncertainty itself chills innovation by exposing developers to unknown risks but giving them no certain way to avoid liability. Justice Stevens, joined by three other members of the Court, recognized the importance of clear and stable guideposts to patent law in his *Bilski* concurrence, noting that “In the area of patents, it is especially important that the law remain stable and clear.” 130 S.Ct. at 3231.

The Federal Circuit standard also interferes with the protected commons that is so vital to the free software community. Injecting patent risk into the commons may wither this free resource from which we all benefit. Most free software developers, only loosely associated with one another, donate their time and skills to further free software’s public purpose, without desire for compensation. If contributing to a free software project exposes the contributor to uncertain risk, fewer will contribute, producing less incremental improvement and innovation for the public’s benefit.

Finally, those volunteers who actually make the

software are often separated by many intermediaries from those who make money off of it, making patent searches unaffordable for most developers. Thus the “deliberate indifference” standard presents free software developers with a choice: face uncertain patent liability for producing source code, or do not contribute at all. This will serve to hold free software projects at a marked disadvantage to their counterparts who produce proprietary software, diminishing their ability to offer their innovations free of charge, impeding progress and increasing the cost of software for all.

CONCLUSION

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

Respectfully submitted,

EBEN MOGLEN

Counsel of record

DANIEL RAVICHER

AARON WILLIAMSON

Software Freedom Law Center

1995 Broadway, 17th Floor

New York, New York 10023

moglen@softwarefreedom.org

212-461-1900

December 6, 2010