

Why the U.S. Supreme Court should protect software innovation

JUNE 4, 2014

FORTUNE



By David J. Kappos
Corporate Partner
Cravath, Swaine & Moore LLP

In a matter of weeks, the U.S. Supreme Court is expected to rule in *Alice Corp. v. CLS Bank International*, a case that should concern any company investing in software innovation. The question before the Court is whether software used in business is eligible to be protected by patent laws. The Patent Act has been interpreted to exclude from patentability, among other things, “abstract ideas.” But U.S. law has historically been interpreted to permit the patenting of software generally.

CLS Bank centers on a process that lessens settlement risk for trades of financial instruments—specifically, whether the process is too abstract for patenting. It was a case that the Federal Circuit, America’s exclusive venue for reviewing patent appeals below the Supreme Court, struggled mightily to decide. That court was unable to reach a consensus view. Six separate opinions were issued, leaving scant instruction for determining when an invention impermissibly has claimed an abstract idea.

The oral arguments suggested that delineating the boundaries of unpatentable abstraction would be no less formidable for the Supreme Court. Justice Stephen Breyer compared the task to that of Odysseus charting his ship’s course between Scylla and Charybdis, two oceanic perils, neither unavoidable, but either of which spelled disaster if approached too closely. He elaborated:

“[I]f you simply say, take an idea that’s abstract and implement it on a computer... if that’s good enough, there is a risk that you will take business in the United States or large segments and instead of having competition on price, service and better production methods, we’ll have competition on who has the best patent lawyer. ... And if you go the other way and say never, then what you do is you rule out real inventions with computers.”

The patenting waters have long been made treacherous by a judicial obsession with the threshold issue of divining categorically those subjects of human creativity eligible for patenting. The issue has incited no fewer than five Supreme Court cases in the last half-century. Whether the Court will succeed this time in providing a map by which the lower courts can navigate these waters remains to be seen. But one thing is certain: Software patents are not, as a class, “bad patents”—that is, too abstract or undeserving. And they should not be curtailed merely on account of our challenges in delimiting precise boundaries for them.

Software patents are more than just code—a point emphasized in the Supreme Court’s oral arguments. Significant technological advancements, from voice recognition to video compression, have been embodied in software—just as those advancements have been implemented at various points also in hardware and firmware, and patented without controversy.

Why should an important technological advance be patentable when implemented in hardware but not in software? Software code, like the English language, is a medium of expression—and no one would think to pass a law that said “no patents in English.” Whatever the relative merits of the claimed invention at the center of *CLS Bank*, the means of its implementation should not affect patentability.

Notably, Justice Sotomayor asked the question outright in *CLS Bank* and neither litigant urged that the Court decide patentability of software generally in order to resolve the case at hand. The U.S. Solicitor General, however, proposed that the Court *could* do so—and *should* do so, at least by nullifying software patents not directed specifically to improvements in “computing technology.” This proposal, however, would merely shift the debate from defining “abstract idea” to defining “computing technology.” In a world where computers are so integrated with other technologies, the question of whether a claimed software-implemented invention improves “computer technology” is not easily answered. Chief Justice Roberts expressed doubts that such an approach would bring about greater clarity and certainty.

Much of the debate in the courts and among America’s innovators is concerned with establishing bright-line rules for patentability of software. Indeed, such rules would bring about greater predictability—a short-term gain for those uncomfortable with a flexible patent regime. But in the long term, inflexible rules would stifle innovation, since today’s rigid decrees cannot possibly account for tomorrow’s as-yet-unknown advancements in technology. The goal cannot be to articulate a watertight definition precisely separating that which, across the entire reach of human creativity, is worthy of patent protection and that which is not.

The worry that too low a threshold for subject-matter eligibility can only lead to the proliferation of bad patents granting unworthy exclusivity is misplaced. Other important requirements for patentability—novelty, non-obviousness, and the requirement that inventors adequately disclose what they are patenting—function as better gatekeepers. Some are hoping for a clear directive from the Supreme Court on subject-matter eligibility. Yet the reality for the extraordinarily nuanced and precariously balanced U.S. intellectual property system is that some level of imprecision (read: flexibility) is not only inevitable but is in fact a source of strength, encouraging innovators to constantly push into new frontiers.

David J. Kappos is a partner at New York City-based law firm, Cravath, Swaine & Moore, where he supports the firm’s clients with a wide range of intellectual property issues. From August 2009 to January 2013, Kappos served as director of the U.S. Patent and Trademark Office, advising the president, secretary of commerce and the administration on intellectual property policy matters.

Reprinted with permission from Fortune.com