

<http://www.scotusblog.com/2014/06/symposium-supreme-court-leaves-patent-protection-for-software-innovation-intact/>

Symposium: Supreme Court leaves patent protection for software innovation intact

David Kappos is a partner at Cravath, Swaine & Moore LLP and the former Director of the U.S. Patent and Trademark Office. He is a senior adviser to the Partnership for American Innovation.

From the perspective of the parties involved, this week's [*Alice Corp. v. CLS Bank*](#) decision held that a process that lessens settlement risk for trades of financial instruments is too abstract for patenting. However, to the leagues of interested onlookers holding their collective breath across our country and indeed around the world, the Supreme Court's unanimous ruling subtly conveyed a much more significant judgment: software, as a class, is every bit as worthy of patent protection as any other medium in which innovation can be practiced.

Despite the loud and sometimes shrill calls from various quarters to curtail "software patenting," the decision is deafening in its declination to do so. The Justices clearly chose not to condemn software to the realm of abstraction, or otherwise express any manner of discrimination against software patents as a class. Instead the Court simply followed its ruling in [*Bilski v. Kappos*](#): to be patent-eligible, an innovation must be patent-worthy in and of itself – regardless of the language in which it is conveyed. In so doing, the Court maintained fidelity to the purpose of the U.S. patent system – that patentability is awarded based on the merits of the invention at hand, whether the medium in which it is built is a concrete structure, ceramic shape, metallic apparatus, molecular composition or computer language.

This week's decision reaffirms that from the point of view of the patent system, software languages are no different from other vernaculars – they are a medium of expression that can be used to capture and convey ideas. The distinction between patentable software in [*Diamond v. Diehr*](#) and unpatentable software in *Bilski* and *CLS Bank* is not about software at all; rather, the difference is the presence or absence of a definitive invention versus abstraction. *Diehr's* new and useful process for curing rubber was held to be innately patentable – the fact that it happened to be manifest in a software language was tributary.

Wisely, rather than labor through a lengthy discussion about software, the Court focused on the abstract nature of Alice Corporation's claimed "invention," declining to delineate a wooden rule for what constitutes unpatentable abstraction. Revisiting a conundrum that has confronted the Court numerous times in the last fifty years, this decision recognizes that rigid boundaries regarding innovation are simply inappropriate. The goal cannot be to articulate a one-size-fits-all definition precisely separating that which, across the entire reach of human creativity, is worthy of patent protection from that which is not. No one is flattered by one-size-fits-all. Flexibility is invaluable in our patent law precisely because it permits necessary adaptation within a system that must continuously evolve to remain vital at the ever-advancing intersection between ingenuity and the future. Inflexible rules inevitably stifle innovation and pick technology winners and losers in blind bets destined for regret. Today's rigid decrees cannot possibly account for tomorrow's as-yet-unknown advances. Thankfully, the Supreme Court showed deep sensitivity to these issues.

What we can account for today is that the most valuable advancements in technology are increasingly reduced to practice in the language of software. Powering the latest technologies in every field, from business services to automobile manufacturing to chemical production to medicines and medical diagnostics, U.S. software innovation is among our most important economic drivers. The U.S. software market not only has grown well above \$150 billion annually, but also has created immense value in all other industries that rely on software — which is to say virtually every industry that exists. Software enables companies across a wide range of industries to create all manner of modern products — from cars to advanced polymers to medicines and medical devices. Companies rely on software to enable their largest endeavors; those that create the most jobs and drive our nation's economy forward. It is thus not surprising that some U.S. states have made software piracy the basis of their efforts to combat international trade-secret misappropriation using unfair competition laws. Massachusetts's 2012 watershed action against Thai-based Narong Seafood Company, for example, demonstrates that software use is vital even in Asian fish markets.

As software has increasingly become a core driver of competitive advantage in every corner of the business world, it is clear that maintaining appropriate incentives and protections for software-based technological advancement is critical to innovation in every sector of the U.S. economy. We are living in the Age of Software, as the Supreme Court's decision in *Alice Corp.* bears testament.

