

## INTELLECTUAL PROPERTY LAW

ADVANCING INNOVATION SINCE 1992

### U.S. SUPREME COURT ADDRESSES ELIGIBILITY OF COMPUTER-IMPLEMENTED PATENTS

On June 19, 2014 the U.S. Supreme Court rendered a decision that will directly affect software related patents. While the ruling addressed a particular case, it will influence numerous decisions in the lower courts as well as the Patent Office. The case, entitled ALICE CORP. PTY. LTD. V. CLS BANK INTERNATIONAL, ET AL., is a patent infringement suit wherein Alice Corporation's ("Alice Corp") patents claimed: (1) a method of exchanging obligations; (2) a computer system configured to carry out the above method; and (3) a computer-readable medium containing program code for performing the method. All of the claims were implemented using a computer. CLS Bank International ("CLS Bank") operated a global network incorporating such methods to facilitate currency transactions. CLS Bank brought suit against Alice Corp, seeking a declaration that the patents were invalid; Alice Corp counter-sued for infringement.

Although the Court found it unnecessary to define the "precise contours of the 'abstract ideas' category," it unanimously held that Alice Corp's claims were each drawn to an "abstract idea." The Court held that there is a two-step approach for assessing whether a claim is directed to patent ineligible subject matter. First, the Court must determine if the claim in issue is directed to one of the implicit exceptions from patent-eligible subject matter (i.e. laws of nature, natural phenomena, or abstract ideas). If so, the Court then asks if the claim elements, considered both individually and "as an ordered combination," transform the nature of the claim into a patent-eligible application.

Applying this approach, the Court stated that the method claims were directed to the abstract idea of intermediated settlement, i.e. using a third party (in this instance, the computer) to mitigate settlement risk. The Court then found that the method claims did no more than simply provide an instruction to implement the abstract idea of intermediated settlement on a generic computer, stating the invention did not improve the functionality of the computer or effect an

improvement in any other technology or technical field. Based on this reasoning, the method claims failed to transform the abstract idea into a patent-eligible invention. The remaining claims were found to recite a handful of generic computer components implementing the same idea. The Alice Corp patents were ruled invalid.

The Court's ruling is consistent with recent public backlash against certain non-practicing entities, or "Patent Trolls". A Patent Troll is a person or company who does not actually manufacture or supply products based upon the patents they own, but rather enforces the patents against accused infringers in an attempt to collect licensing fees. The ruling, however, could impact the Trolls' targets as well as the Trolls themselves. Though the Court speaks in general terms about what constitutes patentable subject matter, it does not provide specific limitations or examples to define it. Thus, some of the same technology companies who have been frequently targeted by Patent Trolls need to be concerned that their own computer-implemented patents may be brought into question.

Only time will tell how the Patent Office will rule on future patent applications. It is assured that if the invention employs a computer-implemented process, the Patent Office will critically scrutinize the patent application. It is our opinion that should the inventor lack a tangible product during the patent prosecution stage, the inventor could be viewed as a Troll by the Examiner, which could affect the subjective reasoning of the Examiner during prosecution. The lower courts will also have various interpretations of this ruling as they apply it in other cases dealing with patent validity of computer-implemented methods. In summary, inventors must now expect greater scrutiny of patents and patent applications in the U.S. for software/computer-implemented methods.

**By Michael Slavin & Brian Taillon**