



## Enfish and chums

James Muraff and Kevin Cukierski of Neal, Gerber & Eisenberg examine a post-Alice Corp patent decision, finding the Court of Appeals of the Federal Circuit well on its way to dealing with the abstract ideas problem

The Supreme Court's 2014 decision in *Alice Corp v CLS Bank* shook up the software patent landscape. Alice introduced a significant roadblock for software developers trying to patent software that functions using only a computer processor, a computer memory, and typical computer input/output devices such as a keyboard and a display. Those seeking to protect software innovations or enforce software patents often have their claims challenged as merely reciting unpatentable abstract ideas—and lose more often than not.

After two years of uncertainty in the software patent world post-Alice, the US Court of Appeals for the Federal Circuit recently opened up a route through the Alice roadblock for certain software innovations. In *Enfish v Microsoft* on 12 May this year, the Federal Circuit found two software patents non-abstract (and thus patent eligible) because they claimed an improvement in computer functionality, and reversed the California district court's summary judgement to the contrary. In clearing up some misconceptions about Alice's reach, the Federal Circuit stated that software patents are not inherently abstract or automatically subject to additional scrutiny under Alice.

*Enfish* clarifies how courts and the US Patent and Trademark Office (USPTO) should apply the two-step patent eligibility test set forth in Alice. The first step of the Alice test asks “whether the claims [of the patent] at issue are directed to a patent-ineligible concept”, such as an abstract idea. If not, the claims are patent eligible. But if the claims are directed to a patent-ineligible concept, one moves on to the second step of the Alice test. The second step asks whether the claims include “significantly more” than the patent-ineligible concept to “transform the ... claim[s]’ into a patent-eligible application” of the patent-ineligible concept. If so, the claims are patent eligible, and vice-versa.

Unfortunately, the Supreme Court never explained how to determine what constitutes an abstract idea sufficient to satisfy the first step of the Alice test. This has caused lower courts (like the district court in the *Enfish* case) and the USPTO to often trivialise the first step of the Alice test by simply: (i) describing the claimed invention as a very broad concept untethered from the actual claim language, (ii) deeming that broad concept an abstract idea, and (iii) moving on to the second step of the Alice test. In many district court and USPTO decisions since Alice, this oversimplification of the claims has more often than not led to negative results for software patent owners and software developers trying to enforce and protect their software innovations.

The Federal Circuit admonished this approach in *Enfish*. The court first put to rest any argument that software patents are by definition abstract in stating that Alice does not hold “that claims directed to software, as opposed to hardware, are inherently abstract and therefore only properly analysed at the second step of the Alice analysis.” Courts and the USPTO must carefully consider the first step of the Alice test for software patents, because “[s]oftware can make non-abstract improvements to computer technology just as hardware improvements can.” It's not the throwaway step that some courts and USPTO patent examiners believe it to be.

The patents at issue in *Enfish* were directed to a self-referential computer database. In reviewing these patents, the Federal Circuit determined that the claims were directed to an improvement in computer functionality, not economic or other tasks for which a computer is used in its ordinary capacity. Specifically, the court found that the claimed self-referential computer database invention improves standard relational databases by enabling faster data searching, more efficient data storage, more flexible database configurations, and faster on-the-fly database software launching.

The court held that this specific improvement to the way computers operate was not an abstract idea, and found the claims patent eligible. Contrast this with the Federal Circuit's decision (on 17 May in *TLI Comm v AV Automotive*, although the court also recognised that tying the claimed software steps to a generic telephone unit and a generic server isn't enough to render the claims patent eligible) that claims including generalised software steps that don't solve a technical problem are directed to an abstract idea.

In *Enfish*, the Federal Circuit also recognised that tying the software to a general-purpose computer did not "doom" the claims to being abstract. In fact, the court clarified that a claim need not define an improvement in computer technology "by reference to 'physical' components" to be patent eligible. "To hold otherwise risks ... creating a categorical ban on software patents."

*Enfish* confirms that software patents aren't a lost cause. District courts and the USPTO are already taking notice and using *Enfish* to reject challengers' attempts to invalidate software patents.

Here are four ways you can use *Enfish* to your benefit to protect your software innovations and rebut Alice arguments.

### Software claims aren't inherently abstract

We've found that some patent examiners believe all software claims are inherently abstract post-Alice. These patent examiners typically gloss over the first step of the Alice test to look for an inventive concept in the second step. A few extreme patent examiners even believe Alice marked the end of software patents.

*Enfish* unequivocally rebuts each of these beliefs. When faced with a patent examiner who believes all software-type claims are inherently abstract, point out that the Federal Circuit found in *Enfish* that "no reason to conclude that all claims directed to improvements in computer-related technology, including those directed to software, are abstract and necessarily analysed at the second step of Alice". And if a patent examiner doesn't think software is patent eligible, educate him or her on the Federal Circuit's finding that "[s]oftware can make non-abstract improvements to computer technology just as hardware improvements can".

### Software functionality can confer patent eligibility

Some patent examiners don't think that software functionality can confer patent eligibility. These patent examiners require the invention to be implemented in hardware other than a generic computer or, in extreme cases, novel hardware. Rebut an argument that software functionality can't confer patent eligibility by pointing out the Federal Circuit's belief that claims are not "doomed" to be ineligible if an "improvement is not defined by reference to

'physical' components". The Federal Circuit acknowledged that requiring a tie to particular hardware would "resurrect a bright-line machine or transformation test" or worse, "creat[e] a categorical ban on software patents."

### Examiners can't oversimplify the claimed invention in the first step of the Alice test

When conducting the first step of the Alice test, many patent examiners trivialise the claims by condensing them into a broad concept—usually a three- or four-word phrase—tenuously linked to the claim language. They then deem that broad concept abstract, and move on to the second step of the Alice test, typically without mention of most of the actual claim language.

This isn't proper anymore (if it ever was in the first place). In *Enfish*, the Federal Circuit criticised the district court because it "oversimplified" the claims and "downplayed the invention's benefits". Rather than trying to condense the claim to a simple phrase, the Federal Circuit focused on the specific claim elements themselves. Part of the reason the court found the claims non-abstract was because the claims weren't "simply directed to any form of storing tabular data, but instead are specifically directed to a self-referential table for a computer database". So, when faced with a patent examiner who is trying to trivialise the claims, point out that the Federal Circuit prohibits that and focus on the actual, specific claim language.

### Claimed inventions that improve technology aren't abstract

Pre-*Enfish*, patent examiners typically wouldn't consider arguments about how the claimed invention improves technology in the first step of the Alice test. They only considered those arguments relevant to the second step of the Alice test. That's no longer the case. When faced with an Alice rejection, argue that the claims are not abstract because the claimed invention improves the relevant technology. A claimed invention that is "directed to a specific improvement to the way computers operate" or that focuses on a "specific asserted improvement" in technology is not abstract, and is patent eligible.

To arm yourself, when drafting a software-based patent application, make sure to discuss the problems in the prior art and explain how the invention solves these problems using computer-related technology, functionality, or capabilities. Point out how the invention improves on existing, conventional technology. Describe the benefits of the invention as compared to the prior art. Even disparage the prior art, if appropriate, such as by discussing its pitfalls (which the invention doesn't have). The Federal Circuit took all of these into account when determining whether the claimed invention in *Enfish* was abstract. **IPPro**

**James Muraff**  
Partner  
Neal, Gerber & Eisenberg LLP



**Kevin Cukierski**  
Associate  
Neal, Gerber & Eisenberg LLP

