TSIRCOU | INTELLECTUAL PROPERTY LAW

Google v. Oracle: The Software Developer's Roller-coaster Ride With No Clear Winner

Oracle v. Google: Software Developer's Roller-coaster Ride

- A crucial holding for software developers is the fact that API's are copyrightable
- However, developers must be vigilant with how they allow free-use of their API's: free-use can mean all use is "Fair Use"

WWW.TSIRCOULAW.COM
515 S. Flower Street, 36th Floor
Los Angeles, CA 90071

May 2016 saw the conclusion of one of the most contentious copyright battles involving software, between two of the industry's heavyweights: Google v. Oracle. The main contention of this lawsuit was the fact that Google had used parts of the Java programming language (owned by Oracle) in their development of the Android mobile phone platform (developed by Google). What made this case so interesting was the calling into question of software development practices that had been considered legitimate for the past two decades, and what made it so important was the potential fallout of a verdict on either side of the issue.

To get a better picture of the issue you must first become a little more acquainted with the history of the case, and the process of software development. Interoperability between software programs, the ability for one program to communicate with another, has long been a core tenant of modern software functionality. Think of your iPhone, you have a number of different apps all created by a number of different developers, however they all still have the ability to communicate seamlessly with each other, the phone, and other associated Apple software. This is because they are all written in the same programming language, using the same set routines, protocols, and tools for building software applications, normally referred to as an API (Application Program Interface). Java is one of the most popular API's used for writing software, so when Google was developing its Android platform it wanted to ensure interoperability with all software written using Java. To accomplish this, Google entered into negotiations with Sun Microsystems (later acquired by Oracle) to incorporate the Java API into their Android system, however negotiations broke down. As a result, Google decided to develop their own version of Java from scratch. While Google's version did not use any of the code from Sun Microsystem's (later Oracle's) version, it did use functions with the same names and functionality to ensure interoperability with Java. This was the basis of Oracle's copyright infringement claim. And so started the software developer's roller-coaster ride.

API’s are copyrightable, but most of their use is “Fair Use”…

The first twist in this case was the ruling at the appeals level that the API code was copyrightable. The API had been deemed solely functional and non-copyrightable in lower courts, however in 2014 the Federal Court of Appeals sided with Oracle, holding the list of Java functions which Google appropriated were in fact copyrightable. This meant that Google was liable for infringement of said copyright, unless they could succeed with an affirmative defense, which leads to the second twist in the case.

While Google was initially dealt a blow with the holding of API’s being copyrightable, it was ultimately successful in arguing their particular use of this API was covered under the doctrine of Fair Use. It is still a matter of hot contention over legal scholars whether or not the jury reached the appropriate conclusion holding Google’s use of the API was covered under Fair Use, with most experts falling on the side of surprise at the jury’s ruling. However, the result still stands; API’s are copyrightable but there is a large amount of lee-way with protected use under Fair Use.

The crux of this case comes back to the point made in the opening about “interoperability” and software development. The past two decades of software development have proceeded with the belief that incorporating parts of an existing API code into your software to allow for interoperability with other software or platforms was an allowable process. This ruling however makes for two linked issues but on the opposite side of the same coin. First, API’s are now decisively covered under copyright law. This means that any owner of an API now has legal recourse to demand royalties from anyone utilizing their code, even if it is just similar functionalities and names for functions. A seeming win for API developers, and a potential nightmare for developers trying to utilize said API’s.

The flipside of the coin however is that Google circumvented this new copyright protection of APIs with a very thin Fair Use argument, even though much of their use of Oracle’s Java API was clearly commercial (to the tune of $9 billion according to Oracle). Many API’s have existed using a dual-licensing scheme: free and open-source license for individual or non-commercial use, and a paying license for commercial utilization. Java followed this model, allowing for free open source licensing on non-commercial uses; allows developers to create software, grow the user base, increasing adoption. When said developer wanted to commercialize the software, they entered into negotiations with the API owner to secure the proper license (like Google tried doing before negotiations broke down). Google ‘s narrative in its Fair Use argument however was that because parts of the Java API were free, all use of it should be free. The fact that this narrative was part of Google’s winning argument should be cause for concern over software developers attempting to utilize, and monetize, this dual licensing structure.

This was a roller-coaster ride with out any clear winners. Of course Google is the immediate winner, not being liable for $9 billion in claimed copyright damages. However looking ahead, the fact that API’s are copyrightable combined with the fact that use, which looks very much commercial in nature, can still be under the ambit of Fair Use, only makes the waters of software copyright litigation that much murkier.