

Google v. Oracle: SCOTUS Grants Cert In The “Copyright Lawsuit of the Decade”—Now What?

December 23, 2019 – The TMCA

by Connor J. Hansen and Stefan Szpajda

Related People

- Connor J. Hansen

Related Capabilities

- Trademark, Copyright + Advertising
- Intellectual Property Litigation



On November 15, 2019, the Supreme Court granted cert in *Google LLC v. Oracle America Inc.* For many observers, this was a long time coming; the parties have been litigating the underlying case since August 2010, and from its inception *Google v. Oracle* has been closely watched by commentators, earning the epithet of “copyright lawsuit of the decade.” By granting cert, the Court has agreed to consider two questions, each of which has the potential to transform copyright law and the software industry.

This post is the first in a new series from the TMCA intended to help make sense of this long-running and multifaceted dispute. Over the next few months, we will publish updates on *Google v. Oracle*, reporting on developments and digging into the key issues implicated by the Court’s forthcoming decision.

Questions Presented

The two questions presented to the Court rest on—and have the potential to materially affect—several key doctrines of copyright law. Of more immediate concern to most people, however, *Google v. Oracle* risks disrupting the software industry. Per Google, if

the Court sides with Oracle on either question, it will prevent developers from adapting software interfaces to build new computer programs, forcing programmers to write new code each time. Per Oracle, if the Court sides with Google, then innovators will no longer be able to protect their investments in software if it becomes popular enough that developers find it commercially advantageous to use it freely.

Scope of Copyright Protection

The first question before the Court is “[w]hether copyright protection extends to a software interface.” Notably, even the scope of this question is in dispute, with Oracle arguing that Google “invented for its petition” the term “software interfaces” to avoid well-settled law that recognizes software as copyrightable. Google responds that Oracle misrepresents the dispute by suggesting it is about software generally, when the question posed to the Court is in fact narrower: whether software interfaces—i.e., what Google alleges are purely functional methods of operating software, as distinct from expressive aspects of the software itself—are copyrightable.

Fair Use

The second question is “[w]hether, as the jury found, petitioner’s use of a software interface in the context of creating a new computer program constitutes fair use.” Apart from the merits, the Court must decide whether the Federal Circuit properly revisited and reversed the jury’s finding of fair use. As to the merits, although fair use is determined based on a four-part test, here the two most important factors were the first and fourth—i.e., the purpose and character of the use, and the effect on existing and potential markets.

This first factor turns on whether Google’s use of Oracle’s code in creating the Android platform is sufficiently “transformative” as that term is used in the fair use analysis. Google argues that software interfaces should be provided “thin” copyright protection given their functional nature and that creating an entirely new mobile platform comprised of relatively few lines of Oracle’s code is an “undoubtedly transformative” use. Oracle responds by stressing that Google’s use is not transformative because Google’s platform uses thousands of lines of Oracle’s code in exactly the same way Oracle uses the code.

The fourth factor turns on whether Google’s use of Oracle source code to build Android harmed Oracle’s market for mobile devices. Google argues that the jury correctly found that Google and Oracle occupy different markets, and by relying on copyright law to create barriers to entry, Oracle “would effectively block competing platforms from accessing developers trained in the Java language.” Oracle responds that Google’s use inflicted “incalculable market harm” on Oracle, and that a use that results in such harm cannot be considered fair use.

What Is This Case Really About?

In brief, the dispute centers around Oracle’s popular Java programming language, which Sun Microsystems originally developed and which Oracle acquired shortly before suing

Google in 2010. The dispute can be summed up as concerning whether the declaring code in Java's application programming interfaces ("API") are **functional**, and thus exempt from copyright protection, or whether they are **expressive**, and thus protected under copyright law. And, per the second question before the Court, if the Java APIs are protected, the extent to which competitors can rely on the doctrine of fair use when using APIs to build new programs.

To better understand the dispute, it helps to have a basic grounding in the structure of the source code at issue. Each API includes hundreds of lines of pre-written code bundled as a package. APIs are designed to perform specific functions when called to do so by a programmer. Programmers use "declaring code" to call one of the pre-written APIs. The declaring code is paired with a corresponding "implementing code" that performs the function of the activated API. For example, if a programmer wishes to display an image on a screen, he or she can simply enter the declaring code for generating a display and the corresponding implementing code will cause the program to display the image without further input from the programmer. APIs are intended to promote efficiency—i.e., a programmer can use a pre-programmed API instead of having to code a function from scratch each time it is used.

Because Google copied only Oracle's declaring code, Google relies on the **functional** nature of APIs to argue that they are exempt from copyright protection. Google analogizes declaring code to a letter on a keyboard, arguing that "[j]ust as a typist writes 'a' by pressing a particular key . . . a developer triggers a particular function by using the relevant declaration to run the corresponding implementing code" and in this way "the Java API facilitates the creation of programs in the Java language across different platforms, much as the now-standard QWERTY keyboard layout facilitates the creation of documents by enabling more efficient typing regardless of the specific word-processing program being used."

As Oracle stresses in its briefing, however, its declaring code did not fall out of the sky. Engineers wrote Oracle's APIs over a period of years at the cost of hundreds of millions of dollars; Oracle thus insists the APIs are the result of **expressive** choices made by those engineers, which entitles APIs to copyright protection. According to Oracle, declaring code is "far more expressive" than the letters on a keyboard, and "communicates to programmers what each program does, how it relates to other programs, and what you need to do to make it work." In contrast, a single key-stroke on a keyboard expresses only the letter itself.

Industry Impact

Although Google's appeal has not yet reached the merits phase, several companies, industry groups, legal scholars, and organizations filed amicus briefs during the cert phase that highlight the legal and practical implications this case will likely have on the software industry. Of the 16 amicus briefs filed so far, three were primarily directed to the copyrightability question, two were directed to the fair use question, and 11 were directed to both questions. Each of the amici supported Google's request to grant cert, except for the Solicitor General of the United States, who argued that both questions were fully and

fairly considered below and the petition should be denied. Although the briefing schedule on the merits phase of Google's appeal remains to be set, Google has requested until January 6, 2020, to file its opening brief. Given the stakes, we expect to see many more amicus briefs filed with the Court at the merits phase.

How Did We Get Here?

One of the reasons *Google v. Oracle* is poised to make such an impact is that it has been around for a long time, and several district court and Federal Circuit opinions have elaborated on the concepts introduced above.

So how did we get here? *Google v. Oracle* arose from one of the most important technological developments of our era: smartphones. When it released Android in 2008, Google wanted developers to be able to quickly and easily develop applications for Android. One way in which Google achieved this goal is by drafting thousands of their pre-written packages of code to perform common functions. Although Google developed its own implementing code to execute each function, it used declaring codes from 37 Java API libraries—which Oracle reminds the Court amounts to 11,500 lines of code—that it considered particularly well-suited to the mobile environment instead of writing its own declaring code for these tasks.

In 2010, Oracle sued Google for patent and copyright infringement, alleging that Google used verbatim copies of Oracle's declaring code from its popular Java coding language to develop Google's Android platform. Over the last decade, this case has bounced back and forth between the United States District Court for the Northern District of California and the United States Court of Appeals for the Federal Circuit.

In 2012, Judge Alsup of the Northern District of California found that declaring code is not protectable because it amounts to a method of operating the Java API, which is exempt from protection under 17 U.S.C. § 102(b). Judge Alsup also found that declaring code is not protectable under the merger doctrine, which provides that no one can claim ownership of an expression "when there is only one (or only a few) ways to express something." Oracle appealed, and the Federal Circuit decided that the declaring code is an expression of an idea that is protectable despite being embodied in a method of operation and that the merger doctrine does not preclude copyright infringement because Oracle had "unlimited options" as to the selection and arrangement of the code that Google copied. Dissatisfied with the Federal Circuit's decision, Google filed a petition for cert at the Supreme Court in 2015. The Solicitor General recommended that the Supreme Court deny cert because the fair use issue had not been fully litigated below. The Supreme Court did not grant cert.

On remand to the Northern District of California in 2016, the jury found that Google's use of the declaring code was a fair use of Oracle's copyright-protected API. Oracle appealed the decision to the Federal Circuit. In the Federal Circuit's 2018 decision, the court again reversed and remanded to the Northern District of California. The Federal Circuit held that fair use did not shield Google's use of the declaring code, primarily because it serves the same purpose and function in Google's Android platform as in Oracle's Java

platform and therefor Google's use is not transformative. The court also held that Google's use is commercial in nature and has caused harm to Oracle because some of Oracle's customers in the tablet and smart-device industry switched from Java to Android. Google again petitioned the Supreme Court, which this time granted cert.

What's Next?

As even this introductory post shows, there is a *lot* to cover here. In the coming months we'll focus on specific issues introduced above to examine how the Court's decision in *Google v. Oracle* could affect copyright law and the software industry.