Proskauer

Landmark Fair Use Victory at the Supreme Court in Software Case

New Media and Technology Law Blog on **April 9, 2021**

In a narrowly drawn, yet significant decision, the Supreme Court reversed the Federal Circuit and ruled that Google LLC's ("Google") copying of some of the Sun Java Application Programming Interface (API) declaring code was a fair use as a matter of law, ending Oracle America Inc.'s ("Oracle") infringement claims over Google's use of portions of the Java API code in the Android mobile platform. (*Google LLC v. Oracle America, Inc.*, No. 18-956, 593 U.S. ___ (Apr. 5, 2021)). In reversing the 2018 Federal Circuit decision that found Google's use of the Java API packages was not fair use, the Supreme Court, in a 6-2 decision (Justice Barrett did not take part in the case) found where Google reimplemented the Java user interface, taking only what was needed to allow outside developers to work in a new and transformative mobile smartphone program, Google's copying of the Sun Java API was a fair use as a matter of law. This decade-long dispute had been previously dubbed "The World Series of IP cases" by the trial court judge, and like many classic series, this one culminated in a winner-take-all Game 7 at the highest court.

Oracle is one of the most notable Supreme Court decisions affecting the software and technology industry in recent memory since, perhaps, the Court's 2010 *Bilski* patent opinion, its 2012 *Jones* decision on GPS tracking, privacy and the Fourth Amendment and its 2005 *Grokster* decision on copyright inducement in the peer-to-peer network context, and certainly the most notable decision implicating fair use since its well-cited 1994 *Campbell* decision that expounded on the nature of "transformative" use. It was no surprise that this case attracted a stack of amicus briefs from various technology companies, organizations, and academia. In the months following oral argument, it was difficult to discern how the Court would decide the case – would it be on procedural grounds based on the Federal Circuit's standard of review of the jury verdict on fair use, on the issue of the copyrightability of the Java API packages, directly on the fair use issue, or some combination. The majority decision is a huge victory for the idea that fair use in the software context is not only a legal defense but a beneficial method to foster innovation by developing something transformative in a new environment on top of the functional building blocks that came before. One has to think hard to recall an opinion involving software and technology that referenced and applied the big picture principles of copyright – "to stimulate artistic creativity for the general public good," as the Supreme Court once stated in a prior case – so indelibly into the fair use analysis.

The decision is also notable for the potential impact on copyright's "transformative use test." By considering Google's intent for using the Java API code, the Court's discussion of what constitutes a "transformative" use appears to diverge somewhat from recent Circuit Court holdings outside the software context. The decision may redirect the transformative use analysis going forward, or future decisions may cabin the holding to the software context.

Recap of the Dispute

In 2010, soon after acquiring Sun Microsystems, Oracle brought suit and alleged that Google infringed the declaring code of certain Java API packages for use in the Android mobile smartphone platform, including copying the elaborate taxonomy covering 37 packages that involves multiple classes and methods. As outlined by the Court, APIs are tools that allow programmers to use prewritten code to build certain functions into their own programs instead of writing such functions from scratch. In short, an API allows programmers to call upon prewritten computing tasks for use in their own programs. APIs contain a "declaring code" and an "implementing code." The declaring code is the language programmers enter to call upon a particular function. The declaring code both labels the particular tasks in the API and organizes those tasks, or "methods," into "packages" and "classes" (or what courts have analogized as file cabinets, drawers, and files). The "implementing code" is the more complex set of code which actually instructs the computer on how to perform the required function. (For example, the declaring code "java.lang.Math.max" corresponds to a more complex set of implementing code which instructs the computer on how to calculate the higher of two numbers.)

During the development of Android, Google had declined to obtain a license from Oracle to use the Java APIs in its platform or license the same under an open source GPL license; instead it copied the declaring code from the 37 Java API packages (over 11,000 lines of code), but wrote its own implementing code that would perform the tasks on a mobile platform. Google designed it this way, believing that Java application programmers would want to find the same 37 sets of functionalities in the new Android system callable by the same names as used in Java. The thinking was that if the Android platform was free and open to use and had a familiar computing language and set of tools, more and more developers would write applications for the platform, which would make Android more attractive to consumers.

The case wended its way through two jury trials, multiple lower court rulings and two trips to the Federal Circuit (Oracle's complaint previously included patent claims). Over the years, the parties wrestled over not only whether the declaring code and the structure, sequence, and organization of the Java API packages are entitled to copyright protection, but also whether Google's unauthorized copying of the declaring code and structure of the 37 Java API packages to use with Google's own original implementing code for its Android operating system constituted fair use.

At the first trial in 2012, a jury ruled that Google infringed Oracle's copyrights in the Java platform, but deadlocked on fair use (the lower court subsequently ruled that the API packages were not copyrightable and entered judgment for Google). However, in 2014, the Federal Circuit reversed and found that the API declaring code was subject to copyright, ruling that a set of commands to instruct a computer to carry out desired operations may contain expression that is eligible for copyright protection and that because the declaring code could have been written and organized in any number of ways and still have achieved the same functions, Section 102(b) of the Copyright Act did not bar the API packages from copyright protection just because they also perform functions. The Federal Circuit remanded the fair use question to the lower court.

At a second trial, the jury ruled that Google's use of the declaring lines of code and the structure, sequence, and organization of the 37 API packages constituted fair use. In a noteworthy ruling, the Federal Circuit overturned the jury verdict and ruled, as a matter of law, that Google's use of the Java API packages was not fair use, and remanded for a trial on damages. (*Oracle America, Inc. v. Google LLC*, 886 F.3d 1179 (Fed. Cir. 2018)). The appeals court held that because the APIs were being used to perform the same tasks, Google's copying and use of the API packages was not "transformative" ("There is nothing fair about taking a copyrighted work verbatim and using it for the same purpose and function as the original in a competing platform"). For additional information, see our prior coverage of this decision.

No Decision on Copyrightability

On appeal, the Supreme Court elected to avoid the first of the two issues presented by the case, whether copyright protection extends to a software interface. In deciding the case on fair use grounds, Justice Breyer wrote "[g]iven the rapidly changing technological, economic, and business-related circumstances, we believe we should not answer more than is necessary to resolve the parties' dispute. We shall assume, but purely for argument's sake, that the entire Sun Java API falls within the definition of that which can be copyrighted." Justice Thomas, in his dissent joined by Justice Alito, would have found Oracle's API code copyrightable (and Google's copying as not fair use). Even without deciding the issue, the majority repeatedly stated that the API code at issue was "inherently bound together with uncopyrightable ideas" and hinted, in dicta, that such functional API declaring code was likely protected by a thin copyright, if any (as Justice Breyer stated, the declaring code was "if copyrightable at all, further than are most computer programs (such as the implementing code) from the core of copyright").

Fair Use Analysis

The second issue in the case is whether, as the jury found, petitioner's use of a software interface in the context of creating a new computer program constitutes fair use.

In their respective briefs, the parties traded richly woven arguments about the copyright and fair use issues as well as the nature and applicability of copyright itself when it comes to the development of new software that reuses existing functional code.

Google: "[As to fair use], the evidence supported the conclusion that Google's reuse of a subset of the Java SE declarations, combined with Google's own vast implementing code, to create an innovative smartphone operating system, was transformative."

Oracle: "Google copied Oracle's code to create a nontransformative derivative: a sequel that adapted Oracle's software for an improved generation of devices. Congress granted Oracle alone the right to create or license such a sequel. Just because Google had the resources to crank out the sequel more quickly does not make it fair."

Google: "Here, the new work is the Android platform, which undoubtedly added something new to the computing world, with a further purpose: It enabled Java developers to unleash their creativity on a new and widely adopted (smartphone) platform, which they could not do while using Oracle's copyrighted work (Java SE)."

Oracle: “Anyone is free to create and organize their own platform that appeals to developers—including one that provides exactly the same functions. Anyone can create a package of programs organized around security functions, or a class of related programs for authenticating data. They simply cannot duplicate Oracle’s organization. The code and organization Google copied are protected because they are expression, not ideas.”

In answering the question of whether Google’s copying and use of the declaring code and organizational structure for 37 Java API was fair use, the Court’s opinion delved into all four of the fair use factors (17 U.S.C. §101) listed below, finding that each factor weighed in favor of fair use.

(1) the purpose and character of the use;

(2) the nature of the copyrighted work;

(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market for or value of the copyrighted work.

As to factor one, the Court considered whether Google’s purpose for using the Java API was “transformative or merely supersedes the objects of the original creation.” Acknowledging that Google did use the Java API to perform the same task it was designed for, the Court reasoned that defining the purpose so narrowly would “severely limit the scope of fair use” in the software context. This is because “virtually any unauthorized use of a copyrighted program” is done for the purpose of performing the program’s original function. Instead, the Court took a broader view of Google’s purpose for using the API. By reimplementing the API on a mobile platform, Google maintained consistency so that programmers could use their acquired skills in a different context – a purpose that the Court found was transformative and “consistent with that creative ‘progress’ that is the basic constitutional objective of copyright itself.” The Court noted that the record demonstrated numerous ways in which reimplementing a functional interface can further the development of computer programs and allow the market to grow.

This portion of the ruling declared that the open source Android platform was transformative in that it built out the Java API in new ways to run in the smartphone environment, which itself had its own novel constraints that weren't present on desktop computers (e.g., power management, network connection issues, GPS capabilities). The decision overturned the Federal Circuit's characterization of Android's reuse of the Java API code as non-transformative and that court's cloistered view that moving material to a new context was not transformative in this instance. Moreover, while the Federal Circuit leaned on Google's commercial purpose in reusing portions of the Java API code in its ruling, the Supreme Court stated that although a lack of commercial use tips the scales in favor of fair use, the "inverse is not necessarily true, as many common fair uses are indisputably commercial." Thus, in finding the first factor weighed in favor of fair use, the commercial nature of the Android project was not dispositive of the first factor, "particularly in light of the inherently transformative role that the reimplementation played in the new Android system."

As to the nature of the copyrighted work, Justice Breyer discussed the functional nature of the Java API code, a consideration that cast a shadow over the remaining factors. The Court explained how the API code was different from many other types of copyrightable computer code, as it is "inextricably" bound together with the uncopyrightable functions and computing tasks that organize tasks into methods and classes as well as new creative expression (Android's own implementing code), and unlike many other programs, its value mainly accrues when third party computer programmers invest time to learn the API's system and continue to use Sun-related implementing programs that Google did not copy. Further, because the copied declaring codes perform a largely organizational rather than expressive or creative purpose, the Court found they are further from the core of copyright than most computer programs (including implementing codes, which Google did not copy). Thus, the Court found this factor weighed in favor of fair use. Interestingly, the Court actually presented this factor first, as the functional nature of the API packages influenced the entire fair use analysis.

As to the third factor, the amount and substantiality of the portion used in relation to the copyrighted work as a whole, the Court held that the “substantiality” factor will generally weigh in favor of fair use where, as here, “the amount of copying was tethered to a valid, and transformative, purpose” (which, in this case, was to not just to make the Java programming language usable on Android systems, but to “permit programmers to make use of their knowledge and experience using the Sun Java API when they wrote new programs for smartphones with the Android platform”). Google had copied around 11,500 lines of Java API code for Android’s codebase, representing 37 separate APIs needed to call up hundreds of different tasks; yet, this was only 0.4% of the total set of Sun Java API computer code, and included only declaring codes – not the implementing codes. The Court determined this was no more than necessary to achieve Google’s objective of allowing programmers to make use of their Java knowledge on the Android platform. Though Google could have created its own, different system of declaring codes, Justice Breyer noted this would not have achieved that basic objective.

Regarding the final fair use factor about market harm, the Court's analysis may have been surprising to some as it was not merely a determination and balancing of Oracle's lost licensing revenue potential but a deeper look and balancing into perceived public benefits of Google's reimplementation of the Java API code. The Court found that Google's Android smartphone platform was simply not a market substitute for Java SE and that, at the time, the evidence suggested that, at a minimum, it would have been difficult for Sun to enter into the emerging smartphone technology market. As the court reasoned, this fact demonstrates that "rather than just 'repurposing [Sun's] code from larger computers to smaller computers,' Google's Android platform was part of a distinct (and more advanced) market than Java software." The Court went further and stated that this fair use factor, in relevant cases, should take into account the public benefits the copying will likely produce as compared to the dollar amounts likely lost by the copyright holder (taking into account the nature of the source of the loss). In concluding that this factor also favored Google, the Court pointed to evidence in the record which reflected the uncertain nature of Sun's ability to compete in Android's market and the involvement of third- party programmers in creating value for the smartphone market, coupled with the risk of creativity-related harms to the public if Oracle were allowed to use copyright to place a "lock" on the Java API declaring code. Because programmers had already gotten used to Oracle's Java API, enforcing the copyright could stymie creative improvements, new applications, and new uses developed by users who have learned to work with that interface.

> "Given the costs and difficulties of producing alternative APIs with similar appeal to programmers, allowing enforcement here would make of the Sun Java API's declaring code a lock limiting the future creativity of new programs. Oracle alone would hold the key. The result could well prove highly profitable to Oracle (or other firms holding a copyright in computer interfaces). But those profits could well flow from creative improvements, new applications, and new uses developed by users who have learned to work with that interface. To that extent, the lock would interfere with, not further, copyright's basic creativity objectives."

Additional Points

- **Fair use and the standard of review.** Some court watchers thought the Court might avoid deciding the substantive issues in this case altogether by taking issue with the Federal Circuit's standard of review of the jury verdict on fair use, which the appeals court ruled ultimately warranted a de novo review. The Court agreed with the Federal Circuit's standard, finding that the fair use question is a mixed question of fact and law, and while reviewing courts should appropriately defer to the jury's findings of underlying facts, the ultimate question of fair use is a legal question for judges to decide de novo.

- **Ruling may be specific to software interfaces**. Despite being a full-throated application of fair use in this case and the need, in certain cases, to consider the role of fair use in fostering innovation and creative progress, Justice Breyer was mindful of maintaining the bounds of the decision (and perhaps careful to keep the six-justice majority together). Reminiscent of the Court's prior ruling in *Aereo*, which cabined the decision to the specific facts and circumstances of that case, the *Oracle* opinion also contained limiting statements: "We do not overturn or modify our earlier cases involving fair use — cases, for example, that involve 'knockoff' products, journalistic writings, and parodies."

- **Competition and licensing issues**. The Court, when discussing the fourth fair use factor, was clearly concerned about what it considered the potential harm to creative software development if Oracle was permitted to leverage its Java-related copyrights to monopolize standard software interfaces that might otherwise be freely used by other developers as building blocks for emerging technologies. In the future, this decision may prompt more disrupters to use fair use as a shield in releasing new products or services that build off of older functional technologies, or otherwise influence negotiations as some potential licensees may find the value of certain functional code to be devalued by the *Oracle*

- **Fair use and software policy.** As the Supreme Court stated in a prior fair use case: "When technological change has rendered its literal terms ambiguous, the Copyright Act must be construed in light of this basic purpose." In many passages of the majority opinion, the idea of fair use was flexible and took account of changes in technology and the functional nature that is inherent in many computer programs (including the Java API packages) as well as the role played by the fair use and the Copyright Act to foster compatibility and innovation in the marketplace: "[F]air use has an important role to play for computer programs by providing a context-based check that keeps the copyright monopoly afforded to computer programs within its lawful bounds."

Some Final Thoughts

The case is certainly a landmark software and fair use ruling and required reading for everyone in the technology industry or copyright field. At first blush, one might see a narrow decision that is fact-specific to functional API code, and perhaps not as immediately influential to fair use as, say, the Second Circuit's 2015 *Google Books* opinion. Still, the Court's measured statements about how to apply the fair use doctrine in software cases are highly significant and likely to have outsized influence in future disputes.

Copyright practitioners will likely note this decision most for the Court's broad view of the transformative use test. The Court's analysis here focused on the subjective intent of the party using the copyrighted work. This diverges from what had appeared to be a growing trend in the transformative use analysis, in which courts stepped back from subjective intent of the alleged infringer and instead focused on a more objective evaluation of how the work may be reasonably perceived. We noted this trend in our recent coverage of the Second and Ninth Circuit's decisions in *Warhol v. Goldsmith* and *Dr. Seuss v. ComicMix LLC*, respectively.

However, the Court's *Google* decision may not represent a total shift in the transformative use analysis; the Court's holding may reasonably be limited to the software context. Writing in dissent, Justice Thomas argued the dangers of applying the Court's broad definition of "transformative" in other contexts. Justice Thomas noted that under the majority's analysis, a movie studio that converts a book into a film without permission would have engaged in a fair use because it created something that allows others to create new products (film reviews, merchandise, interviews, etc).

As with most seminal decisions from the Supreme Court, the true reach of this decision and its impact on the fair use analysis outside of the software context is yet to be seen and will be determined by future lower courts, commentators, and practitioners.

Going forward, we'll also have to see how the ruling affects the software industry regarding open source and the licensing and development of new technologies and platforms that build off of existing programming (and, in the related online media space, whether the decision will affect the fair use calculus surrounding content aggregation innovations, as reading the *Oracle* decision feels much different from the Second Circuit's fair use analysis in the 2018 *TVEyes* case).

- While the Court perhaps did not fully buy into the view of Google and various amici that the “sky would fall” onto the software industry if the Federal Circuit ruling was allowed to stand, the decision is likely to be viewed as a resounding victory for interoperability, collaborative development, and the freedom of developers to create software that works on multiple platforms by re-using existing functional interfaces.
- In the mind of many amici that penned briefs supporting Google’s appeal, the Federal Circuit misapprehended the difference between software interfaces and programs. Thus, in their view, this decision helps to return the software industry back to its general custom that expected “creative” software to receive strong IP protection but tolerated the reuse of functional code by new developers in the name of speed, innovation and interoperability. It is this type of reuse that undergirds much of the latest technologies, including online commerce, cloud computing, and artificial intelligence.
- The decision serves also as a reminder of the fair use doctrine’s role in advancing the big-picture goals of copyright law, as the Supreme Court once stated, not only to “secure a fair return for an ‘author’s’ creative labor,” but also “stimulate artistic creativity for the general public good.”

Anxiety about the outcome of the copyright issues in this dispute has reverberated around the technology sector for years, with some touting the Federal Circuit's holding as an important protection for copyright holders and others decrying that the decision upended years of standard practice in the software industry and would inhibit the development of new programs and result in higher prices for consumers. Open source software advocates also criticized the prior Federal Circuit ruling as impeding innovation, citing the importance of allowing the reuse of APIs to enable the interoperability of open source programs with existing, proprietary software. It is worth restating that we will only know the reach of the Supreme Court's *Oracle* decision after it has been applied to new situations and technologies, but as Justice Breyer stated, "fair use can play an important role in determining the lawful scope of a computer program copyright," particularly when the expressive and functional features of such code are mixed and where application of the doctrine can prevent "illegitimate harms" to future product development. In the meantime, both established software companies and new developers should keep a close eye on how this decision impacts the court's fair use analysis in future tech-related cases, an area which was already given new direction in the Google Books and Google image-search decisions in the prior decades and has now been given another boost that is certain to influence developers who may see new freedoms when it comes to reimplementing or reusing older functional technology in transformative ways.

View Original

Related Professionals

David A. Munkittrick
Partner

Jeffrey D. Neuburger
Partner