The AIPLA Antitrust News

A Publication of the AIPLA Committee on Antitrust Law **May 2021**

Chair's Corner

We hope all committee members are well. With more COVID-19 vaccines becoming available, we look forward to being able to meet together in person again in the foreseeable future.

The 2021 AIPLA virtual spring meeting will be held from May 10-14, 2021, and will feature a keynote by the Hon. Henry C. "Hank" Johnson, Jr., Chairman of the House Judiciary Committee's Subcommittee on Courts, IP and the Internet, Committee, and presentations from UPSTO and private bar thought leaders. The full program and registration are available here.

Our next committee meeting will take place the week after on Tuesday, May 18 from 1-2 pm EST at this zoom link.

This current newsletter contains two case notes. In the first note, Kristina Gliklad from the Antitrust and Competition Group at Baker Botts LLP, provides an overview and analysis of the Fifth Circuit's recent decision in *Impax v. FTC.* The decision affirms a unanimous Commission opinion finding that the reverse payment settlement agreement between Endo Pharmaceuticals Inc. and Impax Laboratories LLC violated federal antitrust laws. The Fifth Circuit's decision provides additional guidance to district courts evaluating reverse-payment settlements, including how the underlying patent litigation should be evaluated, and what courts should consider when weighing any procompetitive benefits arising from patent litigation settlements. The *Impax* decision demonstrates that Commission findings of fact will be granted substantial deference, which may make it difficult for companies to challenge an agency finding of a less restrictive alternative.

The second note, by Jacob Canter from the Litigation and Privacy & Cybersecurity Groups at Crowell & Moring LLP, provides an overview and analysis of the decade-long dispute between Google LLC and Oracle Am., Inc., that just concluded with a decision from the Supreme Court. The case raised important questions about both the copyrightability of computer programs, and under what conditions another's use of a computer program constitutes infringement. Despite its length, very few questions were firmly resolved, and many issues were raised that implicate both copyright and antitrust law. The case note evaluates the major copyrightability and fair use arguments raised in this dispute.

Our Committee aims to publish this newsletter three times each year. We welcome articles on relevant I.P.-antitrust topics. To contribute, please contact Stephen Larson at Stephen.Larson@knobbe.com.

AIPLA Antitrust Committee

Dina Kallay, Chair
Head of Antitrust, Ericsson
Dina.Kallay@ericsson.com

Lisa Kimmel, Vice Chair
Crowell & Moring LLP
LKimmel@crowell.com

Stephen Larson, Editor
Knobbe Martens Olson & Bear LLP
Stephen.Larson@knobbe.com

Impax Laboratories, Inc. v. FTC: Fifth Circuit Grants Substantial Deference to the FTC in Reverse Payment Settlement Case

Kristina Gliklad[1]

On April 13, 2021, the U.S. Court of Appeals for the Fifth Circuit upheld the Federal Trade Commission's ("FTC" or "Commission") unanimous ruling that the reverse payment settlement agreement between Endo Pharmaceuticals Inc. ("Endo") and Impax Laboratories LLC ("Impax") violated federal antitrust laws.[2] This case marks the first time an appellate court has weighed in on the merits of a reverse payment agreement prosecuted by the FTC since the Supreme Court's 2013 ruling in *FTC v. Actavis*.[3]

Factual Background

It all began in 2006 when Endo, the brand-name pharmaceutical company in this case, began selling the only extended-release formulation of oxymorphone (an opioid) called Opana ER. In 2007, Impax filed the first application to market generic extended release oxymorphone. With two of its patents for Opana ER in force until September 2013, Endo sued Impax in January 2008 for patent infringement. Under the Hatch-Waxman Act, this would delay any FDA approval of the generic for thirty months (until June 2010) unless the litigation concluded earlier in Impax's favor.[4]

Separately, Endo was planning to move customers to a new reformulated version of Opana ER that would be protected by new patents and not be therapeutically equivalent to Impax's generic—thereby precluding pharmacists from automatically substituting the generic in place of the brand when filling prescriptions—otherwise known as a "product hop." But the success of Endo's product hop depended on the reformulated Opana ER getting to market sufficiently in advance of Impax's generic product.

With the possible launch date of Impax's generic imminent, Endo and Impax settled the patent litigation in June 2010, shortly after the patent infringement trial began and less than a week before the FDA granted final approval to Impax's generic product. Under the settlement, Impax agreed to delay launching its generic until January 2013—two and half years after it otherwise could have entered at risk, and several months before certain patents for Opana ER expired. In return, Endo agreed not to market its own generic version of extended release oxymorphone until after Impax's six-month

---

[1] Kristina Gliklad is an associate in Baker Botts's Antitrust and Competition Practice in Washington, DC.

[2] *Impax Laboratories, Inc. v. FTC*, --- F.3d ----, 2021 WL 1376984 (5th Cir. Apr. 13, 2021).

[3] *FTC v. Actavis*, 570 U.S. 136 (2013).

[4] Under the Hatch-Waxman Act, a generic drug company can save time and money by filing an Abbreviated New Drug Application with the Food and Drug Administration ("FDA"); and if the generic drug is biologically equivalent to a brand drug the FDA has already approved, then the generic can piggy-back off the pioneer's approval efforts. *See* 21 U.S.C. § 355(j)(2)(A)(i)–(iv) (2018). If, however, the brand manufacturer asserts a patent infringement claim against the generic, then the FDA is stayed from approving the generic application until either thirty months have passed or the patent litigation concludes. 21 U.S.C. § 355(j)(5)(B)(iii).

Hatch-Waxman generic exclusivity period expired in July 2013.[5]

The agreement also required Endo to provide Impax a credit if sales revenues for the original formulation of Opana ER fell by more than fifty percent; this served as insurance in the event Endo transitioned its customers to the reformulated Opana ER between the dates of settlement and Impax's entry—which Endo did in March 2012, resulting in a $102 million credit to Impax. The settlement also included a $10-40 million independent development and co-promotion agreement related to a different drug. Notably, however, Endo's product hop quickly failed when it had to take its reformulated Opana ER off the market due to safety concerns.

Procedural history

In January 2017, the FTC brought separate actions against Endo and Impax, alleging that the reverse payment settlement agreement was both an unfair method of competition in violation of Section 5 of the FTC Act and an unreasonable restraint of trade under the Sherman Act. Endo settled, but Impax decided to litigate the matter. Specifically, the FTC charged Impax with antitrust violations for accepting over $100 million from Endo to delay entry of its generic drug for more than two years. After a three-week trial, the FTC's Administrative Law Judge ("ALJ") ruled for Impax. The ALJ weighed the entirety of the settlement, including the section on independent development and co-promotion of a separate drug, and concluded that the reverse payment was lawful because "as a whole" its procompetitive benefits outweighed the anticompetitive effects; by allowing Impax to enter the market before the Opana ER patents would have expired, consumers benefitted from "uninterrupted and continuous access to generic Opana ER."

In a unanimous decision on April 3, 2019, the full Commission reversed the ALJ. The Commission found that the benefits of the reverse payment agreement must be linked directly to the restraint of competition to outweigh the proof that the restraint harms competition. The ALJ therefore erred by including the development and co-promotion agreement in its analysis because those terms did not provide a justification for the reverse payment itself. Instead, the Commission found that Impax's procompetitive justifications failed here because there were less restrictive ways of achieving the purported benefits. In its appeal, Impax argued that all procompetitive benefits of the settlement should have been weighed in a rule of reason analysis, not just those that justify the reverse payment and entry delay directly.

Fifth Circuit Decision

The Fifth Circuit upheld the FTC's Order and found the Commission had "substantial evidence" to conclude that Endo and Impax entered into an unlawful "pay-for-delay" agreement that replaced the

[5] The Act offers the first generic applicant a six-month exclusivity period during which time the newly approved generic will only compete against the brand drug or generic sold by the brand manufacturer. *See* 21 U.S.C. § 355(j)(5)(B)(iv).

"possibility of competition with the certainty of none."[6]

In its review of the FTC's decision, the court asserted that "'findings of the Commission as to the facts, if supported by evidence, shall be conclusive'"[7] and granted significant deference to the Commission's factual findings. The court explained that this standard required it to go as far as to accept FTC findings supported by substantial evidence "even if suggested alternative conclusions may be equally or even more reasonable and persuasive."[8] And while the court reviewed legal conclusions *de novo*, it nonetheless deferred to the FTC's "informed judgment that a particular commercial practice is to be condemned as unfair."[9]

First, the court agreed that the FTC met its burden to show that the settlement was anticompetitive based on the "large and unjustified" payments from Endo to Impax. Relying on *Actavis,* the Fifth Circuit explained that "the likelihood of a reverse payment bringing about anticompetitive effects depends upon its size, its scale in relation to the payor's anticipated future litigation costs, its independence from other services for which it might represent payment, and the lack of any other convincing justification."[10] Two values instructed the court's reasoning: (1) Endo committed not to market an authorized generic, which increased Impax's projected profits by $24.5 million; and (2) Endo's $102 million payment to Impax as a credit for executing its product hop to the reformulated Opana ER. The panel found that the approximately $3 million Endo saved in litigation expenses from settling was not enough to justify the over $100 million that Endo would end up paying Impax under this agreement, nor did it represent the value of the parties' collaboration to develop a different drug. Instead, the court inferred that at least a portion of that payment must have been for exclusion beyond the point that would have resulted from simply litigating the case to its conclusion—and that the payment's objective was therefore to share supra-competitive prices among the patentee and challenger.

Next, the court rejected Impax's chief argument that the rule of reason required the FTC to do more to balance the anticompetitive effects against the procompetitive benefits. Specifically, Impax contended that the FTC should have looked at the strength of the patents and assessed whether Impax could have entered the market earlier absent the settlement. The court held that *Actavis* does not require Impax's proposed analysis, and the fact that generic competition was "possible," combined with the large payment, was enough to infer anticompetitive effect.

The court further disagreed with Impax that, in hindsight, the settlement does not look anticompetitive. Impax contended that Endo had since obtained more patents for

---

[6] *Impax*, 2021 WL 1376984 at *7.

[7] *Id*. at *4 (quoting 15 U.S.C. § 45(c) (2018)).

[8] *Impax*, 2021 WL 1376984 at *4 (internal quotations omitted).

[9] *Id.* (internal quotations omitted).

[10] *Id.* at *6 (quoting *Actavis*, 570 U.S. at 159).

Opana ER, proven their validity in court, and failed to hop consumers to its reformulated Opana ER, causing Impax's generic to remain as the only version of Opana ER on the market. But the court dismissed these points on the grounds that the competitiveness of the agreement should be measured at the time it was adopted only, thus barring any *ex post* assessment.

Ultimately, the court fully accepted the FTC's findings regarding the viability of less restrictive alternatives and concluded that the anticompetitive effects outweighed any procompetitive benefits. Impax argued that the procompetitive benefits should be measured in the context of the settlement agreement as a whole, thereby obviating the need to show a direct nexus between the monetary payment and the procompetitive benefits. The court did not address whether such a direct nexus existed because the FTC assumed *arguendo* this direct link did exist and nonetheless established through "industry practice, economic analysis, [and] expert testimony" that Impax could have obtained the procompetitive benefits without the reverse payment for delayed entry. Impax's failure to challenge the ALJ's original determination "that a large reverse payment helped induce settlement or that the payment was linked to the January 2013 entry date"[11] further buttressed the court's conclusion that its agreement with Endo was in fact a payment for delayed entry, not a payment for services. In the end, the panel held that the Commission's evidence on the less restrictive alternative was "more than enough" to "allow a reasonable factfinder to conclude that a no-payment settlement was feasible."[12]

Implications

The Fifth Circuit's decision handed the FTC a decisive victory in this long-drawn-out battle against reverse payment settlements. Many lower courts have issued inconsistent opinions as to what agreements constitute a payment, how the underlying patent litigation should be evaluated, and what courts should consider when weighing any procompetitive benefits arising from patent litigation settlements. The *Impax* decision demonstrates that FTC findings of fact will be granted substantial deference, and this may make it difficult for companies to challenge an agency finding of a less restrictive alternative. The decision may also make it harder for manufacturers to defend patent litigation settlements challenged by the FTC containing arguable reverse payments by pointing to other provisions that are procompetitive.

While the Fifth Circuit relied on the Supreme Court's language in *Actavis* permitting "the size of the unexplained reverse payment" to serve as a "workable surrogate for a patent's weakness," certain commentators have raised concerns over the panel's interpretation of the guidance set out in *Actavis*—that only "large and unjustified payments" should be subject to antitrust scrutiny. Such an interpretation may go beyond what *Actavis* and other courts have held by effectively treating the payment itself as the restraint regardless of the strength of the patents at issue or the parties' valuations of the patent infringement action.

The decision does not invalidate Impax's agreements with Endo or impose

[11] *Impax*, 2021 WL 1376984 at *6.

[12] *Id.* at *12.

any monetary sanctions; instead, the court's decision leaves in place the FTC's cease-and-desist order enjoining Impax from entering similar reverse payment settlements going forward. With growing calls for antitrust reform, as well as the FTC's recently announced pharmaceutical mergers working group, these and other conduct challenges in the pharmaceutical sector are unlikely to leave the spotlight anytime soon.

The Supreme Court Speaks in *Google v. Oracle*

Jacob Canter
Crowell & Moring LLP[13]

The ten-year battle between Oracle America, Inc. ("Oracle") and Google LLC ("Google") is finally over. On April 5, 2021, the United States Supreme Court held that Google's copying of a portion of Oracle's Java code constituted a fair use and thus did not violate the copyright law.[14] In reaching this decision, the Supreme Court "assume[d], for argument's sake, that the [relevant code] was copyrightable," but did not actually address this question.[15]

The decision may have important implications for both copyright and antitrust law for years to come. It may also impact how the software technology market where Google and Oracle both compete develops and changes – particularly in regards to software that facilitates "interoperability," or the ability to run other software across different platforms and devices. But before we can analyze the impact of this extraordinary legal dispute, we should make sure everyone understands what was actually at issue and what actually happened. This case note describes the long road from filing to final resolution in *Oracle v Google*, as well as the main competing arguments that the courts grappled with and that we are now left to interpret.

The Technology

Sun Microsystems, Inc. built a popular software programming language called Java. A software programming language is, like any other language, a collection of "words, symbols, and other units, together with syntax rules for using them to create instructions."[16] Java is referred to as a "language" or a "platform" because human programmers can learn Java and write their own applications for any general purpose (just like how I can learn English and write this relatively straightforward article, or George Saunders can learn English and write the much more beautiful, thoughtful, and creative *Lincoln in the Bardo*). Crucially, Java has been so successful and is so valuable because the platform is configured "to run on different types of computer hardware, without having to rewrite [Java applications] for each different type."[17] This is Java's important "interoperability" functionality.[18]

The only relevant Java code for our purposes is 37 Java application programming interface (or "API") packages. An API package is a

---

[13] This article represents the views of the author and not Crowell & Moring LLP or any of its clients.

[14] *See Google LLC v. Oracle Am., Inc.*, 141 S. Ct. 1183 (2021) ("Supreme Court Fair Use Opinion").

[15] *Id*. at 1190.

[16] *Oracle Am., Inc. v. Google., Inc.*, 750 F.3d 1339, 1348 (Fed. Cir. 2014) ("Appellate Copyrightability Opinion").

[17] *Oracle Am., Inc. v. Google, Inc.*, 872 F. Supp. 2d 974, 977 (N.D. Cal. 2012) ("District Copyrightability Opinion").

[18] The courts all grabbed onto Java's marketing phrase "write once, run anywhere" to elucidate the value of the Java platform's interoperability functionality. *See e.g.*, Supreme Court Fair Use Opinion, 141 S. Ct. at 1190.

collection of "ready-to-use"[19] and "pre-written"[20] code that runs basic (but often essential) functions. Java API packages are organized into classes, and then further organized into methods. For example, one Java API package is "java.lang." Within this package is a class called "math." And within the "math" class is a method called "max," which allows the programmer to find the larger of two integers.[21] Thus, rather than having to write many lines of code so that an application can determine the larger of two integers, a Java programmer needs to only write the brief API package for the software to have this functionality. Judge Alsup (N.D. Cal.) explained that an API package is "like a library" where "each package is like a bookshelf . . . each class is like a book . . . [and] each method is like a how-to-do-it chapter in a book. Go to the right shelf, select the right book, and open it to the chapter that covers the work you need."[22]

Importantly, each API package is written with two different types of code, the "declaring code" and the "implementing code." "Declaring code is the expression that identifies the prewritten function."[23] The "main point [of declaring code is to] introduce[] the method body and [to] specif[y] very precisely the inputs, name and other functionality" for the command.[24] In his dissent, Justice Thomas explained that declaring code is like a defined term in a statute, which has a very precise and technical meaning that legislatures can refer to quickly and efficiently through the single word or phrase.[25] The implementing code, on the other hand, "gives the computer the step-by-step instructions for carrying out the declared function," and the "expressions used by the programmer from the declaring code command the computer to execute the

---

[19] Appellate Copyrightability Opinion, 750 F.3d at 1348.

[20] District Copyrightability Opinion, 872 F. Supp. at 977.

[21] *See* Appellate Copyrightability Opinion, 750 F.3d at 1349. Other examples of methods are code to print onto a screen or retrieve the cosign of an angle. Methods can be extraordinarily more complicated as well.

[22] District Copyrightability Opinion, 872 F. Supp. 2d at 977. Justice Breyer offered his own "comprehensive, albeit farfetched" analogy for API packages: "Imagine that you can, via certain keystrokes, instruct a robot to move to a particular file cabinet, to open a certain drawer, and to pick out a specific recipe. With the proper recipe in hand, the robot then moves to your kitchen and gives it to a cook to prepare the dish. This example mirrors the API's task-related organizational system. Through your simple command, the robot locates the right recipe and hands it off to the cook. In the same way, typing in a method call prompts the API to locate the correct implementing code and hand it off to your computer. And importantly, to select the dish that you want for your meal, you do not need to know the recipe's contents, just as a programmer using an API does not need to learn the implementing code. In both situations, learning the simple command is enough." Supreme Court Fair Use Opinion, 141 S. Ct. at 1193.

[23] Appellate Copyrightability Opinion, 750 F.3d at 1349.

[24] District Copyrightability Opinion, 872 F. Supp. 2d at 979-80.

[25] Supreme Court Fair Use Opinion, 141 S. Ct. at 1211 (Thomas, J., dissent).

associated implementing code."[26] In Justice Thomas' analogy, the implementing code is "the detailed definition in the statute" that actually explains what the defined term covers.[27]

The 37 Java API packages equal roughly 11,500 lines of code. This is approximately 0.4% of the total 2.86 million lines of Java API code.[28] It is, however, "virtually all the declaring code needed to call up hundreds of different tasks."[29] Also, the courts appeared to accept that at least a few of these Java API packages are essential or "core" for developing Java applications.[30]

The Legal Questions

Oracle acquired Sun in 2010. By this time, Google had already sought to license the entire Java language for use in developing its Android software platform for mobile devices. But these negotiations failed. After the negotiations failed, "Google copied the declaring source code from the 37 Java API packages verbatim, inserting that code into parts of its Android software. In doing so, Google copied the elaborately organized taxonomy of all the names of methods, classes, interfaces, and packages—the 'overall system of organized names—covering 37 packages, with over six hundred classes, with over six thousand methods.'"[31]

Oracle sued Google for copyright infringement.[32] For our purposes, two copyright theories are relevant. First, Oracle argued that Google infringed the literal declaring code. Second, Oracle argued that Google infringed the "structure, sequence, and organization" of the Java API packages. Google responded that neither the declaring code nor the structure, sequence, and organization ("SSO") of the API packages are copyrightable; even if the declaring code and SSO are copyrightable, Google's use was not infringing; and even if Google's use was infringing, it nonetheless did not violate the Copyright Act because its use constituted fair use.

The Lower Court Copyrightability Decisions

The District Court in the Northern District of California held a six-week trial where the parties presented evidence on copyrightability, infringement, and fair use. The jury was directed to presume copyrightability and reach decisions solely on infringement and fair use, and the Court (post-trial) issued a decision on

---

[26] Appellate Copyrightability Opinion, 750 F.3d at 1349.

[27] Supreme Court Fair Use Opinion, 141 S. Ct. at 1211 (Thomas, J., dissent).

[28] *See* Appellate Copyrightability Opinion, 750 F.3d at 1348 (quoting District Copyrightability Opinion, 872 F. Supp. 2d at 977 n.2).

[29] Supreme Court Fair Use Opinion, 141 S. Ct. at 1204.

[30] *See e.g.*, District Copyrightability Opinion, 872 F. Supp. 2d at 982; Appellate Copyrightability Opinion, at 750 F.3d at 1349; Supreme Court Fair Use Opinion, 141 S. Ct. at 1205.

[31] Appellate Copyrightability Opinion, 750 F.3d at 1351 (quoting District Copyrightability Opinion, 872 F. Supp. 2d at 999).

[32] Oracle also sued for patent infringement, but those claims did not survive the first jury trial and were not further pursued.

copyrightability.[33] The jury found that Google infringed the 37 Java API packages but deadlocked on whether the use was protected by the fair use doctrine.[34]

The District Court held that neither the declaring code nor the SSO of the API packages were copyrightable.

The District Court provided two bases for concluding that the declaring code is not copyrightable. First, it noted that federal copyright law expressly excludes names and short phrases from its protection.[35] It then argued that declaring code is nothing more than names and short phrases.[36] Second, the District Court argued that the declaring code is not copyrightable pursuant to the merger doctrine, which provides that "when there is only one (or only a few) ways to express something, then no one can claim ownership of such expression by copyright."[37] It explained that "[t]o carry out any given function, the method specification as set forth in the declaration *must be identical* under the Java rules (save only for the choices of argument names). Any other declaration would carry out some *other* function. The declaration requires precision. . . . Therefore, there can be no copyright violation in using the identical declarations."[38] The District Court applied the same analysis to the classes: a programmer must write the class precisely under the Java rules to achieve the desired function; therefore, there is no distinction between the *expression* of the class and the *idea* that the class expresses; thus, the merger doctrine precludes classes from copyright protection.[39]

As to the copyrightability of the SSO, the District Court acknowledged that the SSO "resembles a taxonomy" and thus "resembles" a creative means of organizing material (and thus is at least potentially copyrightable).[40] But the District Court stated that the SSO "is *also* a command structure for a system or method of operation of the [API]."[41] This matters because section 102(b) of the Copyright Act provides that "in no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, [or] method of operation."[42] Based on § 102(b), and guided by binding and persuasive precedent, the District Court stated that a taxonomical structure that is functionally necessary to express the underlying idea is not

---

[33] As Judge Alsup explained: "In this way, the court of appeal would have a wider range of alternatives without having to worry about an expensive retrial. Counsel were so informed but not the jury." District Copyrightability Opinion, 872 F. Supp. 2d at 975-76.

[34] *Id*.

[35] *See* District Copyrightability Opinion, 872 F. Supp. 2d at 983-84 (citing 37 C.F.R. 202.1(a)).

[36] *See id*. at 998. The District Court kept this argument very brief. This is understandable given that the opinion is otherwise extensive and extraordinarily detailed.

[37] *Id*. at 998 (emphasis in original). The District Court also noted that because "[e]ach method has a singular purpose or function" each method is an unprotectable process and not a potentially protectable creative combination of processes. *Id*.

[38] *Id*.

[39] *Id*.

[40] *Id*. at 999.

[41] *Id*. (emphasis in original).

[42] *Id*. (quoting 17 U.S.C. § 102(b)).

copyrightable.[43] And the District Court explained that, despite its creative features, the SSO "is nevertheless . . . a long hierarchy of over six thousand commands to carry out pre-assigned functions."[44] Thus, pursuant to § 102(b), the SSO is not copyrightable.

To "shed further light on the character of the command structure as a system or method of operation," the District Court concluded by discussing the concept of interoperability:

> Surely, millions of lines of code had been written in Java before Android arrived. These programs necessarily used the java.package.Class.method() command format. These programs called on all or some of the specific 37 packages at issue and necessarily used the command structure of names at issue. Such code was owned by the developers themselves, not by Oracle. *In order for at least some of this code to run on Android, Google was required to provide the same java.package.Class.method() command system using the same names with the same "taxonomy" and with the same functional specifications.* Google replicated what was necessary to achieve a degree of interoperability —but no more, taking care, as said before, to provide its own implementations.[45]

The Federal Circuit had appellate jurisdiction because the initial case included patent claims.[46] The three-judge panel disagreed with the District Court and held that both the declaring code and the SSO of the API packages were copyrightable. Given both this outcome and that the jury hung on Google's fair use defense, the case was reversed and remanded for a new trial on fair use.[47]

The Appellate Court disagreed with the District Court at basically every point in the analysis.

As to the declaring code, the Appellate Court held that copyrightability is a "low bar" and that none of the District Court's or Google's bases for precluding copyrightability are applicable.[48] The merger doctrine is inapplicable because "Oracle had 'unlimited

[43] *Id*. at 985-997 (where the Court extensively analyzes § 102(b) and related precedent).

[44] *Id*. at 999-1000. The District Court stated this legal rule as follows: "Under Section 102(b), copyright protection never extends to an idea, procedure, process, system, method of operation or concept regardless of form. Functional elements essential for interoperability are not copyrightable." *Id*. at 997.

[45] *Id*. (emphasis in original).

[46] Appellate Copyrightability Opinion, 750 F.3d at 1353 (citing 28 U.S.C. § 1295(a)(1)).

[47] *See id*. at 1376 ("[W]e find that due respect for the limit of our appellate function requires that we remand the fair use question for a new trial."). Google also filed a writ of certiorari to the Supreme Court, but the writ was denied. *See Google, Inc. v. Oracle Am., Inc.*, 135 S. Ct. 2887 (2015) (Mem.).

[48] Appellate Copyrightability Opinion, 750 F.3d at 1354.

options as to the selection and arrangement of the 7000 lines Google copied.'"[49] The names and short phrases exception is inapplicable because "the original *combination*" of the copied declaring code was creatively put together by Sun, even if each particular name or phrase is not copyrightable.[50] And the scenes a faire doctrine (which the District Court rejected and Google nonetheless raised on appeal) is inapplicable because the doctrine only "excludes from protection against infringement expression whose creation flowed naturally from considerations external to the author's creativity,'" and Google offered no evidence to support this theory.[51]

As to the SSO, the Appellate Court held that the District Court improperly applied the "method of operation" standard for determining copyrightability, which provides that any expression that is part of a necessary functional operation cannot be copyrighted.[52] "The problem with [this] approach," the Appellate Court explained, "is that computer programs are by definition functional—they are designed to accomplish some task."[53] Thus, the panel reasoned that "[i]f we were to accept the district court's suggestion that a computer program is uncopyrightable simply because it 'carries out pre-assigned functions,' no computer program [would be] protectable."[54] Instead, the proper approach is the "abstract-filtration-comparison analysis," whereby "an original work—even one that serves a function—is entitled to copyright protection as long as the author had multiple ways to express the underlying idea."[55] Because "Oracle claims copyright protection only in its *particular* way of naming and organizing each of the 37 Java API packages," and is not claiming copyright "in the idea of organizing functions of a computer program or in the 'package-class-method' organizational structure in the abstract," the SSO is copyrightable.[56] The upshot is that because Google "could have" structured the SSO of Android differently, the SSO that Google did copy was copyrightable. [57]

---

[49] *Id*. at 1361.

[50] *Id*. at 1363.

[51] *Id*. at 1364 (quoting *Miltel, Inc. v. Iqtel, Inc.*, 124 F. 3d 1366, 1375 (10th Cir. 1997)).

[52] *Id*. at 1365-67.

[53] *Id*. at 1367.

[54] *Id*. (the Appellate Court then went on to say: "Though the trial court did add the caveat that it 'does not hold that the structure, sequence[,] and organization of all computer programs may be stolen, [District Copyrightability Opinion], 872 F. Supp. 2d at 1002, it is hard to see how its method of operation analysis could lead to any other conclusion.").

[55] *Id*. The abstract filtration test comes initially from *Computer Associates Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992). Explaining the curious name "abstract filtration" is not actually elucidating.

[56] Appellate Copyrightability Opinion, 750 F.3d at 1367.

[57] *Id*. at 1365; *see also id*. ("Thus, Oracle concedes that Google and others could employ the Java language—much like anyone could employ the English language to write a paragraph without violating the copyrights of other English language writers. And, that Google may employ the "package-class-method" structure much like authors can employ the same rules of grammar chosen by other authors without fear of infringement. What Oracle contends is that, beyond that point, Google, like any author, is not permitted to

Finally, the Appellate Court held that the District Court's analysis of Java's interoperability functionality was improper for three reasons. First, interoperability is "only relevant, if at all, to fair use—not to the question of whether the API packages are copyrightable."[58] This is because questions about interoperability are really about how the software interacts with other software, and thus boil down to whether the way another uses the software is legally permissible.[59] Second, the "relevant compatibility inquiry asks whether the plaintiff's choices were dictated by a need to ensure that its program work with existing third-party systems," and not, as the District Court asked, "[w]hether a defendant later seeks to make its program interoperable with the plaintiff's program."[60] And finally, the panel simply felt that Google's interoperability argument was "confusing":

> While Google repeatedly cites to the district court's finding that Google had to copy the packages so that an app written in Java could run on Android, it cites to no evidence in the record that any such app exists and points to no Java apps that either pre-dated or post-dated Android that could run on the Android platform. The compatibility Google sought to foster was not with Oracle's Java platform or with the JVM central to that platform. Instead, Google wanted to capitalize on the fact that software developers were already trained and experienced in using the Java API packages at issue. . . . Google's interest was in accelerating its development process by leveraging Java for its existing base of developers.[61]

The Supreme Court Opinions

Google prevailed at the second jury trial. After the jury verdict, Oracle filed a motion for judgment as a matter of law as to the fair use determination, which the District Court denied.[62] The Federal Circuit then reversed the District Court, holding that – even when assuming all factual questions in Google's favor – Google's use of the Java code was in fact not fair as a matter of law.[63]

The Supreme Court then granted certiorari on both copyrightability and fair use, and two

---

employ the precise phrasing or precise structure chosen by Oracle to flesh out the substance of its packages—the details and arrangement of the prose.").

[58] *Id*.

[59] *See id*. at 1368-70.

[60] *Id*. at 1370-71; *id*. at 1371 ("Stated differently, the focus is on compatibility needs and programming choices of the party claiming copyright protection—not the choices the defendant made to achieve compatibility with the plaintiff's program.").

[61] *Id*. at 1371-72 (citation to record omitted).

[62] *Oracle Am. Inc. v. Google, Inc.* 2016 WL 3181206 (N.D. Cal. June 8, 2016) ("District Court Fair Use Opinion"). Oracle also filed a renewed JMOL and moved for a new trial, and both also failed.

[63] *See Oracle Am. Inc. v. Google LLC*, 886 F.3d 1179 (Fed. Cir. 2018) ("Appellate Fair Use Opinion").

opinions were issued.[64] In the majority opinion written by Justice Breyer, the Supreme Court held that Google's use of the Java code was fair, and assumed solely for argument's sake that both the declaring code and SSO are copyrightable. In the dissenting opinion, Justice Thomas (joined by Justice Alito) argued that the declaring code and SSO are both copyrightable and that Google's use should not be considered fair.[65]

The majority first explained why it did not address copyrightability. "A holding for Google on either question presented," the majority stated, "would dispense with Oracle's copyright claims."[66] Thus, "[g]iven the rapidly changing technological, economic, and business-related circumstances," the majority decided to "answer not more than is necessary to resolve the parties' dispute."[67]

Next, the majority articulated its general views about copyright's role – and particularly the fair use doctrine's role – in policing the boundaries of software ownership:

> The upshot, in our view, is that fair use can play an important role in determining the lawful scope of a computer program copyright, such as the copyright at issue here. It can help to distinguish among technologies. It can distinguish between expressive and functional features of computer code where those features are mixed. It can focus on the legitimate need to provide incentives to produce copyrighted material while examining the extent to which yet further protection creates unrelated or illegitimate harms in other markets or to the development of other products. In a word, it can carry out its basic purpose of providing a context-based check that can help to keep a copyright monopoly within its lawful bounds.[68]

Only then did the majority address the four fair use factors.[69] "For expository purposes," it started with factor two – "the nature of the copyrighted work."[70] For two reasons, the majority held that this factor weighs in favor of fair use because "the declaring code is, if copyrightable at all, further than are most computer programs (such as the implementing code) from the core of

---

[64] *See* Supreme Court Fair Use Opinion, 141 S. Ct. at 1195.

[65] The final outcome was 6-2 because Justice Barrett took no part in consideration or decision of the case.

[66] Supreme Court Fair Use Opinion, 141 S. Ct. at 1197.

[67] *Id*.

[68] *Id*. at 1198.

[69] Actually, the majority first clarified that fair use is a mixed question of fact and law and that while reviewing courts should "defer to jury findings on underlying facts, the ultimate question whether those facts showed a 'fair use' is a legal question for judges to decide *de novo*." *Id*. at 1199. It also clarified that this approach does not violate the Seventh Amendment. *Id*. at 1200-01.

[70] *Id*. at 1201-02.

copyright."[71] First, declaring code simply "differs from many other kids of copyrightable computer code":

> [Declaring code] is inextricably bound together with a general system, the division of computing tasks, that no one claims is a proper subject of copyright. It is inextricably bound up with the idea of organizing tasks into what we have called cabinets, drawers, and files, an idea that is also not copyrightable. It is inextricably bound up with the use of specific commands known to programmers, known here as method calls (such as **java.lang.Math.max**, etc.), that Oracle does not here contest. And it is inextricably bound up with implementing code, which is copyrightable but was not copied.[72]

Second, not only does declaring code generally differ from other types of code, but it also "embodies a different type of creativity" than implementing code:

> Sun Java's creators, for example, tried to find declaring code names that would prove intuitively easy to remember. They wanted to attract programmers who would learn the system, help to develop it further, and prove reluctant to use another. Sun's business strategy originally emphasized the importance of using the API to attract programmers. It sought to make the API "open" and "then compete on implementations." The testimony at trial was replete with examples of witnesses drawing this critical line between the user-centered declaratory code and the innovative implementing code.
>
> These features mean that, as part of a user interface, the declaring code differs to some degree from the mine run of computer programs. Like other computer programs, it is functional in nature. But unlike many other programs, its use is inherently bound together with uncopyrightable ideas (general task division and organization) and new creative expression (Android's implementing code). Unlike many other programs, its value in significant part derives from the value that those who do not hold copyrights, namely, computer programmers, invest of their own time and effort to learn the API's system. And unlike many other programs, its value lies in its efforts to encourage

[71] *Id*. at 1202.

[72] *Id*. at 1201 (bold in original). Also, to understand the cabinets, drawers, and files reference, see footnote 9, supra.

> programmers to learn and to use that system so that they will use (and continue to use) Sun-related implementing programs that Google did not copy.[73]

Despite acknowledging that "Google copied portions of the Sun Java API precisely, and [that] it did so in part for the same reason that Sun created those portions," the majority also found that the first factor – the purpose and character of the use – favors Google.[74] By far, the most important reason for this conclusion was that the majority considered Google's use of the API transformative.[75] It stated that because "Google's use of the Sun Java API seeks to create new products [and] seeks to expand the use and usefulness of Android-based smartphones" the use is "consistent with that creative 'progress' that is the basic constitutional objective of copyright itself."[76] And the majority also agreed with Google and several *amici* that using new implementing code with an existing API is a useful way to "further the development of computer programs."[77]

The majority also found that the third factor – the amount and substantiality of the portion used – favors Google.[78] While recognizing that the copied code "amount[ed] to virtually all the declaring code needed to call up hundreds of different tasks," it stated that "[s]everal features of Google's copying suggest that the better way to look at the numbers is to take into account the several million lines that Google did not copy."[79] First, it stated that the API "is inseparably bound to those task-implementing lines. Its purpose is to call them up."[80] Second, because "Google copied those lines not because of their creativity, their beauty, or even (in a sense) because of their purpose. It copied them because programmers had already learned to work with the Sun Java API's system, and it would have been difficult, perhaps prohibitively so, to attract programmers to build its Android smartphone system without them."[81] And third, because "Google's basic purpose was to create a different task-related system for a different computing environment (smartphones) and to create a platform—the Android platform—that would help achieve and popularize that objective."[82]

The majority also disagreed with the Federal Circuit's argument that "Google could have achieved its Java-compatibility objective by

---

[73] *Id*. at 1202 (citations to the Record excluded).

[74] *Id*. at 1202-04.

[75] The majority also noted that the commercial nature of Google's copying did not necessarily tip the scale against fair use, and that the bad faith copying of Oracle's code was potentially not even relevant to the analysis and, in any event, not serious enough to change the outcome. *Id* at 1204.

[76] *Id*. at 1203.

[77] *Id*. at 1203-04. On this point, I note that the amicus brief of the American Antitrust Institute was cited, particularly the following sentence: "Copyright on largely functional elements of software that [have] become an industry standard gives a copyright holder anti-competitive power."

[78] *Id*. at 1204-06.

[79] *Id*. at 1204, 1205.

[80] *Id*. at 1205.

[81] *Id*.

[82] *Id*.

copying only 170 lines of code that are 'necessary to write in the Java language.'"[83] It explained that the Federal Circuit viewed Google's "legitimate objectives too narrowly" and that "Google's basic objective was not simply to make the Java programming language usable on its Android systems. It was 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."[84] It stated that "[i]n a sense the declaring code was the key that it needed to unlock the programmers' creative energies. And [Google] needed those energies to create and to improve its own innovative Android systems."[85]

Finally, the majority also stated that the fourth factor – market effects – favored Google.[86] The majority began by noting that this factor "must take into account the public benefits the copying will likely produce," whether those benefits are "related to copyright's concern for the creative production of new expression," and how those benefits compare "with dollar amounts likely lost (taking into account as well the nature and source of the loss)."[87] It then provided three bases for its determination. First, that the jury was shown substantial evidence that it could reasonably take to mean that Oracle's financial loss due to the copying was not so severe.[88] Second, that the reason why Oracle would have made money through the Java API (*i.e.*, it's first-mover advantage) is not consistent with finding against fair use on this factor.[89] And third, that "given programmers' investment in learning the Sun Java API, to allow enforcement of Oracle's copyright here would risk harm to the public."[90]

Justice Thomas' dissent disagreed on every point with the majority. He first argued that the declaring code is copyrightable. It is copyrightable both because the Copyright Act expressly recognizes that "computer programs" are entitled to copyright protection, and because it clearly meets the "extremely low" bar for copyrightability.[91] Justice Thomas then rejected Google's "method of operation" argument because the declaring code and implementing code are "'inextricably bound' together" and thus there is no reasonable way to distinguish between the two for purposes of copyrightability.[92] And he rejected the

---

[83] *Id*. at 1205 (quoting Appellate Fair Use Opinion, 886 F.3d at 1206).

[84] *Id*.

[85] *Id*. at 1205-06.

[86] *Id*. at 1206-08.

[87] *Id*. at 1206.

[88] *Id*. at 1206-07.

[89] *Id*. at 1207-08 ("It is important, however, to consider why and how Oracle might have become entitled to money. When a new interface, like an API or a spreadsheet program, first comes on the market, it may attract new users because of its expressive qualities, such as a better visual screen or because of its superior functionality. As time passes, however, it may be valuable for a different reason, namely, because users, including programmers, are just used to it. They have already learned how to work with it.").

[90] *Id*. at 1208.

[91] *Id*. at 1212 (Thomas, J., dissenting).

[92] *Id*. at 1213 (quoting the majority opinion at 1201).

merger doctrine argument because "there may have been only one way for Google to copy the lines of declaring code, but there were innumerable ways for Oracle to write them."[93]

Next, Justice Thomas criticized the majority's refusal to engage in the copyrightability question, calling it "far from ordinary."[94] He argued that "Congress rejected categorical distinctions between declaring and implementing code" yet the majority created such a distinction in its opinion.[95] And he stated that "[t]he result of this distorting analysis is an opinion that makes it difficult to imagine any circumstance in which declaring code will remain protected by copyright."[96]

Justice Thomas' view of the four factors differed from the majority as well. On factor two – nature of the copyrighted work – he again stressed that the majority's distinction between declaring and implementing code is untenable under copyright law.[97] He also disputed the majority's argument that declaring code is "inherently bound" with uncopyrightable ideas, noting that many things (like books and the use of plots) meet this criterion.[98] And he stated that it "makes no difference that the value of declaring code depends on how much time third parties invest in learning it" because "[m]any other copyrighted works [such as a Broadway musical script that needs actors and singers] depend on the same."[99]

Justice Thomas next turned to the market effects factor, which he referred to as "[u]ndoubtedly the single most important element of fair use."[100] For two reasons, he argued that this factor strongly favors no fair use, the upshot being that "by copying Oracle's work, Google decimated Oracle's market and created a mobile operating system now in over 2.5 billion actively used devices, earning tens of billions of dollars every year."[101] First, he argued that "Google eliminated the reason manufacturers were willing to pay to install the Java platform."[102] And second, he argued that "Google interfered with opportunities for Oracle to license the Java platform to developers of smartphone operating systems," a basis that he also alleges the majority ignored.[103]

For Justice Thomas, the first factor (purpose and character of the use) strongly favored no fair use as well. He called the commercial nature of Google's use of the copied material

---

[93] *Id*.

[94] *Id*. at 1214.

[95] *Id*.

[96] *Id*.

[97] *Id*. at 1215-16.

[98] *Id*. at 1215.

[99] *Id*. at 1216. Justice Thomas did not appear to make an affirmative argument on factor two.

[100] *Id*. (citing *Harper & Row, Publishers, Inc. v. Nation Enterprises*, 471 U.S. 539, 566 (1985)).

[101] *Id*. at 1218.

[102] *Id*. at 1216.

[103] *Id*. at 1216-17. Justice Thomas also strongly criticized the majority's framework for considering market effects, accusing the majority of failing to consider "what Google *has* done" with its market control when focusing on what Oracle might have done. *Id*. 1217.

"overwhelming."[104] And he rejected that there was anything transformative about Google's use of code "for the exact same purpose Oracle did."[105] And Justice Thomas felt just as strongly about the third factor (amount and substantiality of the portion used), stating that the declaring code which Google copied verbatim was "the heart or focal points of Oracle's work" and "what attracted programmers to the Java platform and why Google was so interested in that code."[106]

Conclusion

The several opinions surveyed here from the *Oracle v Google* saga (and this is in fact not all of them) are astonishing for several reasons. They are astonishing for their length, for the creativity, and for the polar variations in the conclusions reached. For so much thought and attention given to these subjects in this decade of litigation – and for so many productive legal contributions that move our practice forward – very little, it seems, was actually firmly resolved in this dispute. So it goes, perhaps. Or at least onto the next one.

[104] *Id*. at 1218.

[105] *Id*. at 1219. Indeed, Justice Thomas stated that the majority conflated the concepts of transformative works and derivative works.

[106] *Id*. at 1220.