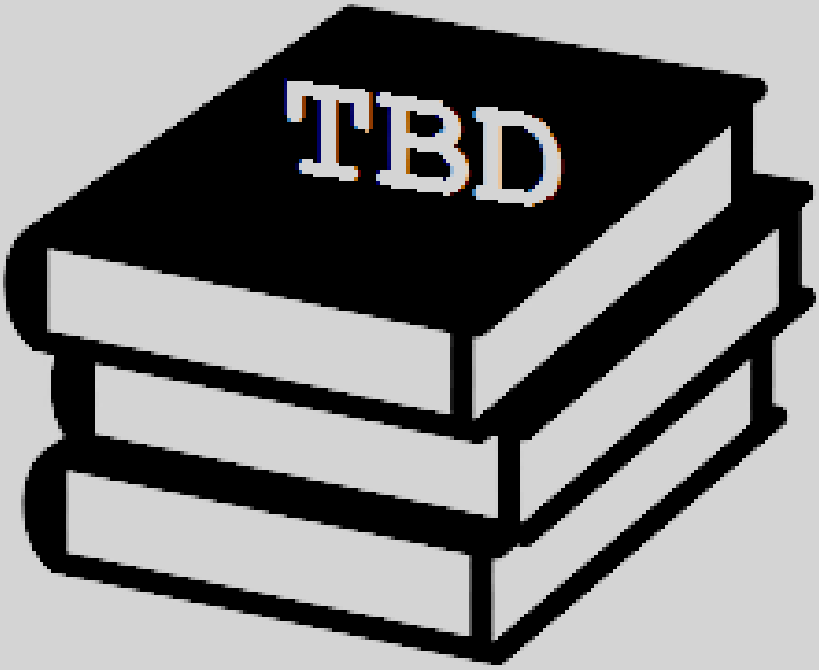
BUY™	SELL™	SHOP™



SELL YOUR OWN SAMPLES

(help others get the justice that they deserve)

BUY™	SELL™	SHOP™

No.

In the Supreme Court of the United States

GOOGLE LLC, PETITIONER

v.

ORACLE AMERICA, INC.

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

PETITION FOR A WRIT OF CERTIORARI

THOMAS C. GOLDSTEIN
GOLDSTEIN & RUSSELL, P.C.
5225 Wisconsin Avenue, N.W., Suite 404
Washington, DC 20015

ROBERT A. VAN NEST
CHRISTA M. ANDERSON
EUGENE M. PAIGE
REID P. MULLEN
KEKER, VAN NEST & PETERS LLP
633 Battery Street
San Francisco, CA 94111

BRUCE W. BABER
MARISA C. MALECK
KING & SPALDING LLP
1180 Peachtree Street, N.E.
Atlanta, GA 30309

KANNON K. SHANMUGAM
Counsel of Record
CHARLES L. MCCLOUD
MENG JIA YANG
WILLIAMS & CONNOLLY LLP
725 Twelfth Street, N.W.
Washington, DC 20005
(202) 434-5000
kshanmugam@wc.com

MICHAEL S. KWUN
KWUN BHANSALI LAZARUS LLP
555 Montgomery Street, Suite 750
San Francisco, CA 94111

QUESTIONS PRESENTED

The Copyright Act provides that, while "original works of authorship" are generally eligible for copyright protection, 17 U.S.C. 102(a), "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work," 17 U.S.C. 102(b). The Act also makes clear that "the fair use of a copyrighted work * * * is not an infringement of copyright." 17 U.S.C. 107.

As is relevant here, software interfaces are lines of computer code that allow developers to operate prewritten libraries of code used to perform particular tasks. Since the earliest days of software development, developers have used interfaces to access essential tools for building new computer programs. Contravening that longstanding practice, the Federal Circuit in this case held both that a software interface is copyrightable and that petitioner's use of a software interface in a new computer program cannot constitute fair use as a matter of law.

The questions presented are:

1. Whether copyright protection extends to a software interface.

2. Whether, as the jury found, petitioner's use of a software interface in the context of creating a new computer program constitutes fair use.

CORPORATE DISCLOSURE STATEMENT

Petitioner Google LLC is an indirect subsidiary of Alphabet Inc., a publicly held company. Alphabet Inc. has no parent corporation, and no publicly held company owns 10% or more of its stock.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases:

Page

Cases—continued:

Page

Cases—continued:

Statutes:

Miscellaneous:

Page

Miscellaneous—continued:

In the Supreme Court of the United States

No.

GOOGLE LLC, PETITIONER

v.

ORACLE AMERICA, INC.

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

PETITION FOR A WRIT OF CERTIORARI

Google LLC respectfully petitions for a writ of certiorari to review the judgment of the United States Court of Appeals for the Federal Circuit in this case.

OPINIONS BELOW

The opinion of the court of appeals regarding fair use (App., *infra*, 1a-55a) is reported at 886 F.3d 1179. The district court's orders denying respondent's motions for judgment as a matter of law (App., *infra*, 92a-120a) and for a new trial (App., *infra*, 56a-91a) are unreported.

The earlier opinion of the court of appeals regarding copyrightability (App., *infra*, 121a-192a) is reported at 750 F.3d 1339. The district court's order granting petitioner's motion for judgment as a matter of law (App., *infra*, 212a-272a) is reported at 872 F. Supp. 2d 974.

JURISDICTION

The judgment of the court of appeals was entered on March 27, 2018. A petition for rehearing was denied on August 28, 2018 (App., *infra*, 299a-300a). On October 23, 2018, the Chief Justice extended the time within which to file a petition for a writ of certiorari to and including January 25, 2019. The jurisdiction of this Court is invoked under 28 U.S.C. 1254(1).

STATUTORY PROVISIONS INVOLVED

The relevant provisions of the Copyright Act of 1976, Pub. L. No. 94-553, 90 Stat. 2541, are reproduced in the appendix to this petition (App., *infra*, 285a-299a).

STATEMENT

This case has been aptly described as the "copyright lawsuit of the decade." Anandashankar Mazumdar, *Oracle Victory Stirs Uncertainties in Software Copyright*, Bloomberg Law (May 10, 2018) <tinyurl.com/suitdecade>. As it comes to this Court, the case presents two exceptionally important questions concerning the copyrightability and fair use of software interfaces—lines of computer code that are necessary to allow developers to operate prewritten libraries of code used to perform particular tasks.

The first question is whether copyright protection extends to a software interface. This Court granted certiorari to resolve a closely related question in *Lotus Development Corp.* v. *Borland International, Inc.*, 516 U.S. 233 (1996) (per curiam), but the Court deadlocked 4-4 on that question after oral argument. The second question 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. The lower courts are badly in

need of guidance on how to apply the fair-use doctrine in the context of computer code.

This case involves a high-profile dispute between two leading technology companies, petitioner Google and respondent Oracle. Sun Microsystems originally developed the Java platform, which includes the free Java programming language. The software interfaces at issue are part of the Java language's application programming interface (API). Sun encouraged developers to learn the Java language by touting the ability to use software interfaces—also known as "the Java API declarations"—to access preexisting libraries of code used to perform particular tasks. The interfaces thereby facilitated development of programs in the Java language.

Google used some of the Java API declarations to build Android, a revolutionary platform for modern mobile devices such as smartphones and tablets. Google incorporated those declarations to allow developers to write applications for Android using the Java language. Sun originally applauded Google for using the Java language. But after Oracle acquired Sun, it sued Google for copyright infringement.

After years of litigation, the Federal Circuit (which had jurisdiction because of Oracle's initial assertion of now-dismissed patent claims) has twice reversed judgments in Google's favor. It first held that the Java API declarations are copyrightable and then overturned a jury's verdict that Google's use of the declarations constituted fair use. The Federal Circuit thereby deepened the acknowledged conflicts among the courts of appeals concerning the application of the Copyright Act and the merger doctrine in the context of computer software.

Google has never disputed that some forms of computer code are entitled to copyright protection. But the Federal Circuit's widely criticized opinions—in an area in

which that court has no specialized expertise—go much further, throwing a devastating one-two punch at the software industry. If allowed to stand, the Federal Circuit's approach will upend the longstanding expectation of software developers that they are free to use existing software interfaces to build new computer programs. Developers who have invested in learning free and open programming languages such as Java will be unable to use those skills to create programs for new platforms—a result that will undermine both competition and innovation. Because this case is an optimal vehicle for addressing the exceptionally important questions presented, the petition for a writ of certiorari should be granted.

A. Background

1. Modern smartphones are "such a pervasive and insistent part of daily life that the proverbial visitor from Mars might conclude they were an important feature of human anatomy." *Riley* v. *California*, 134 S. Ct. 2473, 2484 (2014). Given the ubiquity of smartphones today, it is easy to forget the challenges that developers initially faced in building the operating systems that allow modern smartphones to perform their myriad functions. Among other things, developers had to account for smaller processors, limited memory and battery life, and the need to support mobile communications and interactive applications. C.A. App. 21958.

In 2008, Google overcame those challenges and released Android, an open-source platform designed to enable mobile devices such as smartphones and tablets. The Android platform took over three years to build, and Google had almost 100 engineers working on the project. C.A. App. 21858, 21861-21862. In the decade since its release, Android has become one of the most widely used

mobile operating systems, with billions of users worldwide.

Google built Android using the Java programming language. Sun Microsystems released the Java language in the 1990s and made it free and open for all to use without a license. Sun's motives for doing so were not entirely altruistic: Sun believed that the resulting proliferation of Java developers would drive sales of Sun hardware and other services. Java has become one of the world's most popular programming languages. App., *infra*, 216a.

Java 2 Standard Edition (Java SE) is a platform used to write and run programs for desktop and server computers. Java SE includes the Java language, which in turn contains the Java application programming interface (API). C.A. App. 51447-51448. The Java API provides access to prewritten "methods." In Java, as in many other programming languages, methods are used to program specific, commonly performed tasks. Each method consists of two parts: a method header and a method body. The method header is also known as a "declaration" or "declaring code," because it labels (or "declares") the method, typically by reference to what the method will do. The declaration also includes information about where the method is located in the Java API libraries. The method body, also known as the "implementing code," is the underlying code that actually performs the task stated in the declaration. App., *infra*, 126a.

The relationship between the declaration and the implementing code is analogous to the interaction between a keyboard and a word-processing program. Just as a typist writes "a" by pressing a particular key, causing the word-processing program to display that letter, a developer triggers a particular function by using the relevant declaration to run the corresponding implementing code. By allowing developers easily to access the libraries of

prewritten code in a standard manner, 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.

Once the corresponding declaration is used for a particular method, the developer does not need to worry about or even understand the specifics of the method's implementing code. Instead, all a developer needs to do to invoke a method is to use a shorthand command derived from the method's declaration. App., *infra*, 4a-5a, 126a-127a. By using the shorthand commands, a developer can create complex software without having to write new implementing code for every routine task. *Id.* at 4a.

3. In 2005, Google and Sun began discussing a partnership that would have allowed Google to adapt the entire Java SE platform for smartphones. Google and Sun conducted negotiations but were unable to reach an agreement. In the absence of such an agreement, Google used the freely available Java language (and its declarations) to develop its own libraries of methods that enabled developers to build smartphone applications for use on Android devices. App., *infra*, 106a-107a, 117a, 218a-219a.

At the same time, Google understood that developers would want to use their existing Java language skills to create Android applications, including their knowledge of familiar declarations and shorthand commands to trigger common operations. For those commands to work on the Android platform, Google had to replicate the syntax and structure of the Java API declarations *exactly*; any change to those declarations would have prevented developers from reusing the same commands, thereby forcing them to learn new commands for each routine task. Google accordingly used the same declarations for certain

methods in 37 Java API libraries that were determined by Google to be "key to mobile devices." App., *infra*, 219a. For every one of those methods, however, Google wrote its own implementing code, tailoring the code to accommodate the unique challenges of the smartphone environment. *Id.* at 218a-219a.

Because Google independently wrote the implementing code that formed the body of each method, using only certain declarations, only 3% of the code was the same across the 37 disputed Java API libraries and the corresponding Android libraries. App., *infra*, 220a. In total, that overlapping code represented less than 0.1% of the over 15 million relevant lines of code in Android.

B. Procedural History

1. Sun was aware that Google was developing Android using the Java language, including the API declarations, but never objected or mentioned its Java copyrights to Google. C.A. App. 50363, 51692-51693. To the contrary, Sun initially celebrated the launch of Android. Its chief executive officer publicly offered "heartfelt congratulations" to Google, stating that Google had "strapped another set of rockets to the [Java] community's momentum." *Id.* at 55325.

In 2010, however, Oracle acquired Sun. A few months after the acquisition, Oracle sued Google in the United States District Court for the Northern District of California, alleging seven counts of patent infringement. Oracle eventually withdrew five of its patent-infringement claims, and the jury found against Oracle on the two remaining claims.

Oracle's complaint also asserted a single claim of copyright infringement. Although the Java language was free and open, Oracle claimed that Google's use of the

Java API declarations infringed Oracle's copyrights. Oracle asserted that Google had impermissibly copied the declarations and also the "structure, sequence, and organization" of the Java API; Oracle premised the latter claim on the theory that the declarations "embod[ied] the structure" of the Java API by specifying the name and location of each method. App., *infra*, 140a.

After a two-week trial, the jury considered the copyright-infringement claims but was ultimately unable to reach a verdict, hanging on Google's fair-use defense. The district court then granted Google's motion for judgment as a matter of law on the copyright claims. App., *infra*, 212a-272a. The district court held that the Java API declarations were not copyrightable because they constituted a "method of operation" under 17 U.S.C. 102(b). App., *infra*, 262a-263a, 265a. The court further held that the declarations were not copyrightable under 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." *Id.* at 261a. The court also denied Oracle's motion for judgment as a matter of law on Google's fair-use defense. *Id.* at 211a.

2. The Federal Circuit reversed and remanded. App., *infra*, 121a-192a.

Recognizing a three-way circuit conflict on the copyrightability question, the Federal Circuit first reasoned that the merger doctrine was "irrelevant" to copyrightability and was in any event not satisfied here, because Sun could have written the declarations in more than one way. App., *infra*, 142a-143a, 148a, 150a-151a. The Federal Circuit then reasoned that Section 102(b) "does not extinguish the protection accorded a particular expression of an idea merely because that expression is embodied in a method of operation." *Id.* at 161a (internal quota-

tion marks and citation omitted). In the court's view, Section 102(b) served only to codify the "idea/expression dichotomy"—that is, the principle that "[c]opyright protection extends only to the expression of an idea—not to the underlying idea itself." *Id.* at 137a. The Federal Circuit remanded for a new trial on Google's fair-use defense, concluding that the record did not "contain[] sufficient factual findings upon which [the court] could base a de novo assessment." *Id.* at 184a.

3. Google petitioned for a writ of certiorari, and this Court called for the Solicitor General's views. The government acknowledged that Google's petition raised "substantial and important" concerns about the effects of enforcing Oracle's copyrights on software development, including lock-in effects and restrictions on interoperability. 14-410 U.S. Br. 10, 17. But the government recommended against certiorari, citing the case's then-interlocutory posture and noting that its concerns could be addressed on remand through the fair-use defense. See *id.* at 10, 22. This Court denied review. 135 S. Ct. 2887 (2015).

4. On remand, after another two-week trial featuring dozens of witnesses and hundreds of exhibits, the jury found that Google had engaged in fair use. The district court denied Oracle's motions for judgment as a matter of law and for a new trial. App., *infra*, 56a-120a.

5. The Federal Circuit again reversed and remanded. Having concluded in the first appeal that the fair-use defense should be decided by a jury because the panel could not resolve the underlying factual issues, the same panel reversed course and held that Google had not engaged in fair use as a matter of law. App., *infra*, 1a-55a.

The non-exclusive factors relevant to determining fair use include (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 on the potential market for or value of the copyrighted work. See 17 U.S.C. 107. In holding that Google had not engaged in fair use, the Federal Circuit focused primarily on the first and fourth factors. App., *infra*, 25a-53a.

As to the first factor, the Federal Circuit determined that the commercial nature of Google's use of the declarations weighed against a finding of fair use. App., *infra*, 25a-28a. In considering whether Google's use was transformative, the Federal Circuit asserted that the declarations served the same function in Android as in the Java platform, and it concluded on that basis that the declarations themselves had not been transformed—even though Google used the declarations to create an entirely new smartphone platform and developed new implementing code tailored to the smartphone environment. *Id.* at 28a-37a.

As to the fourth factor, the Federal Circuit found that Java SE had been used in early mobile phones, which meant that "Android competed directly with [Java] in the market for mobile devices." App., *infra*, 50a. And even if Java SE had not been so used, the Federal Circuit would still have concluded that there was market harm by considering how "Google's copying affected potential markets Oracle *might* enter or derivative works it *might* create or license others to create." *Id.* at 51a (emphases added).

Weighing the four enumerated factors together, and without considering other relevant evidence as this Court has required, the Federal Circuit held that Google did not engage in fair use as a matter of law. App., *infra*, 53a-54a. Having overturned the jury's verdict, the court remanded for a trial on damages. *Id.* at 54a-55a.

6. After calling for a response, the Federal Circuit denied Google's petition for rehearing. App., *infra*, 283a-284a.

REASONS FOR GRANTING THE PETITION

The questions presented in this case are of critical importance to the computer software industry, one of the principal drivers of the nation's economy. Because new software builds on components of existing software, innovation in this field largely depends on how copyright law treats software interfaces, the essential building blocks of software development. The Federal Circuit has upended the computer industry's longstanding expectation that developers are free to use software interfaces to build new computer programs. In the opinions under review, the Federal Circuit first deemed software interfaces to be copyrightable, then held that petitioner's reuse of such interfaces could not be fair use as a matter of law because the interfaces performed the same function in the new software.

The Federal Circuit has deepened an existing circuit conflict over the copyrightability of software interfaces. Other courts of appeals have concluded that similar interfaces are not copyrightable under both the plain language of Section 102(b) of the Copyright Act and the merger doctrine. And as to fair use, the Federal Circuit misapplied the doctrine and rendered it essentially impossible for the reuse of software interfaces to qualify as fair use. The Court should review and correct the Federal Circuit's distortion of copyright law in an area crucial to technological innovation.

A. This Court Should Grant Review To Decide Whether Copyright Protection Extends To A Software Interface

The Federal Circuit's first opinion deepens acknowledged conflicts among the courts of appeals regarding the

proper interpretation of Section 102(b) of the Copyright Act and the application of the merger doctrine.

1. Under Section 102(b), copyright protection does not extend to "any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such [original] work." 17 U.S.C. 102(b). More than two decades ago, the Court granted certiorari to consider whether that provision "bars protection for [a] menu command hierarchy despite its expressive characteristics, because it assists users in communicating with a computer program in order to perform useful operations." Br. at i, *Lotus Development Corp.* v. *Borland International, Inc.*, 516 U.S. 233 (1996) (No. 94-2003). The Court deadlocked 4-4 on that question after oral argument, and the division among the courts of appeals has only grown with the intervening rise of the modern software industry. Given its obvious importance and its close relation to the question left unresolved in *Lotus*, the question presented here cries out for the Court's review.

a. The First and Sixth Circuits have held that Section 102(b) precludes copyright protection for all methods of operation, including those embodied in computer software interfaces.

The First Circuit's decision in *Lotus* concerned the menu command hierarchy in Lotus 1-2-3, a then-ubiquitous spreadsheet program. See 49 F.3d 807, 809 (1995). The First Circuit acknowledged that "the Lotus developers made some expressive choices" in creating the hierarchy, but it nevertheless held that the hierarchy constituted a "method[] of operation" and was thus excluded from copyright protection under Section 102(b). *Id.* at 816. That was true regardless of whether the developers "could have designed the Lotus menu command hierarchy

differently." *Ibid.* Because the "menu command hierarchy provides the means by which users control and operate" the spreadsheet program, the hierarchy constituted a method of operation. *Id.* at 815.

The Sixth Circuit adopted a similar rule in *Lexmark International, Inc.* v. *Static Control Components, Inc.*, 387 F.3d 522 (2004). The court reasoned that, "even if a work is in some sense 'original' under § 102(a), it still may not be copyrightable because [of] § 102(b)." *Id.* at 534. The Sixth Circuit reaffirmed that understanding in *ATC Distribution Group, Inc.* v. *Whatever It Takes Transmissions & Parts, Inc.*, 402 F.3d 700 (2005). There, it explained that, although methods of operation may be "[o]riginal and creative," Section 102(b) excludes them from copyright protection because they are "the idea itself" rather than the "expression of the idea." *Id.* at 707.

b. For its part, the Third Circuit has taken the diametrically opposite position, holding that Section 102(b) was "not intended to enlarge or contract the scope of copyright protection" but rather to codify the "somewhat metaphysical" dichotomy between idea and expression. *Apple Computer, Inc.* v. *Franklin Computer Corp.*, 714 F.2d 1240, 1252, 1253 (1983), cert. denied, 464 U.S. 1033 (1984). In the Third Circuit's view, a "method of operation" embodied in a software interface is copyrightable as long as it could have been written differently and still serve the same high-level purpose, such as "to aid in the business operations of a dental laboratory." *Whelan Associates, Inc.* v. *Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1238 (1986), cert. denied, 479 U.S. 1031 (1987).

c. The Second Circuit has adopted still another approach: the so-called "abstraction/filtration/comparison" test. See *Computer Associates International, Inc.* v. *Altai, Inc.*, 982 F.2d 693, 706 (1992). Under that test, a court

should first "dissect the allegedly copied program's structure and isolate each level of abstraction contained within it." *Id.* at 707. The court should then "filter[] * * * protectable expression from non-protectable material." *Ibid.* Finally, after isolating the "golden nugget" of "protectable expression," the court should inquire whether "the defendant copied any aspect of this protected expression." *Id.* at 710. The Second Circuit's test has since been adopted by the Fifth and Tenth Circuits (with the Tenth Circuit expressly rejecting the First Circuit's approach in *Lotus*). See *Engineering Dynamics, Inc.* v. *Structural Software, Inc.*, 26 F.3d 1335, 1342 (5th Cir. 1994); *Engineering Dynamics, Inc.* v. *Structural Software, Inc.*, 46 F.3d 408, 409 (5th Cir. 1995) (supplemental opinion); *Mitel, Inc.* v. *Iqtel, Inc.*, 124 F.3d 1366, 1371-1372 (10th Cir. 1997).

d. In this case, the Federal Circuit purported to apply the Second Circuit's abstraction/filtration/comparison test. At the same time, however, the Federal Circuit also relied on the Third Circuit's more categorical test, which the Second Circuit's test was intended to replace. App., *infra*, 142a-143a, 161a-162a; see *Computer Associates*, 982 F.2d at 705-706.

In short, the courts of appeals are deeply divided on the appropriate standard for determining the circumstances under which a software interface is copyrightable under Section 102(b). At a minimum, the Federal Circuit's standard directly conflicts with the standard adopted by the First and Sixth Circuits. The Court should grant review to resolve the conflict among the courts of appeals on this exceptionally important issue.

2. In addition to deepening a circuit conflict regarding the proper interpretation of Section 102(b), the Federal Circuit took sides in a related circuit conflict concern-

ing the merger doctrine. That doctrine follows from Section 102(b)'s exclusion of ideas from the scope of copyright protection; it provides that, where an idea is incapable of being expressed in more than one way, the idea and expression "merge" and become unprotectable. 4 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* §13.03[B][3] (2015) (Nimmer).

The Federal Circuit concluded that the merger doctrine does not restrict copyright protection for computer code necessary for interoperability as long as the original author could have written the code in more than one way. App., *infra*, 150a-151a. In so concluding, the Federal Circuit aligned itself with the Third Circuit, which has stated that, "once the plaintiff creates a copyrightable work, a defendant's desire 'to achieve total compatibility * * * is a commercial and competitive objective which does not enter into the * * * issue of whether particular ideas and expressions have merged.'" *Id.* at 171a (quoting *Apple Computer*, 714 F.2d at 1253).

The position of the Third and Federal Circuits on the role of interoperability in the merger doctrine is directly contrary to that of the Sixth Circuit, which has found that "[p]rogram code that is strictly necessary to achieve current compatibility presents a merger problem, almost by definition, and is thus excluded from the scope of any copyright." *Lexmark*, 387 F.3d at 536 (internal quotation marks omitted).

The Federal Circuit further concluded that the merger doctrine plays no role in the copyrightability analysis and is instead merely an affirmative defense to *infringement* once copyrightability has been established. App., *infra*, 144a-145a. That position accords with holdings of two circuits, but it cannot be reconciled with the holdings of two others. See *Kregos* v. *Associated Press*, 937 F.2d 700, 705 (2d Cir. 1991), cert. denied, 510 U.S. 1112 (1992);

Ets-Hokin v. *Skyy Spirits, Inc.*, 225 F.3d 1068, 1082 (9th Cir. 2000); but see *Lexmark*, 387 F.3d at 535; *Veeck* v. *Southern Building Code Congress International, Inc.*, 293 F.3d 791, 801-802 (5th Cir. 2002) (en banc), cert. denied, 539 U.S. 969 (2003). That conflict is important in its own right, and it further supports this Court's review here.

3. The Federal Circuit erred in holding that the Java API declarations were copyrightable.

a. To begin with, the declarations constitute uncopyrightable methods of operation. Section 102(b) provides that "in no case does copyright protection for an original work of authorship extend to any * * * method of operation * * * regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. 102(b).

Ignoring the plain text of the statute, the Federal Circuit held that "components of a program that can be characterized as a 'method of operation' may nevertheless be copyrightable." App., *infra*, 161a. According to that court, "Section 102(a) and 102(b) are to be considered collectively so that certain expressions are subject to greater scrutiny." *Id.* at 141a-142a.

The Federal Circuit's approach is untenable, and its bottom line is incorrect. The Java API declarations simply tell developers how to access the prewritten methods to perform tasks carried out by the implementing code. App., *infra*, 4a-5a, 126a-127a. In that respect, the declarations are analogous to a set of rules developers are trained to follow when writing programs in the Java language. If the rules were changed, the prewritten methods would not work. For that reason, the declarations are necessarily part of the method of operating the libraries of prewritten code. See, *e.g.*, *Lotus*, 49 F.3d at 817-818.

The conclusion that the declarations are uncopyrightable is not affected by the fact that other aspects of the Java API libraries, like the implementing code, may be copyrightable. Quite to the contrary, "[t]he mere fact that a work is copyrighted does not mean that every element of the work may be protected." *Feist Publications, Inc.* v. *Rural Telephone Service Co.*, 499 U.S. 340, 348 (1991); see Pamela Samuelson, *Why Copyright Law Excludes Systems and Processes From the Scope of Its Protection*, 85 Tex. L. Rev. 1921, 1921 (2007). In the specific context of computer programs, the legislative history of Section 102(b) shows that Congress intended to "make clear that the expression adopted by the programmer is the copyrightable element in a computer program," while "the actual processes or methods embodied in the program are not within the scope of the copyright law." H.R. Rep. No. 1476, 94th Cong., 2d Sess. 56-57 (1976); S. Rep. No. 473, 94th Cong., 1st Sess. 54 (1975).

Consistent with Congress's expectation, courts have held that "aspects" of a computer program that constitute "functional requirements for compatibility" with other programs are "not protected by copyright" under Section 102(b). *Sega Enterprises Ltd.* v. *Accolade, Inc.*, 977 F.2d 1510, 1522 (9th Cir. 1992); see *Sony Computer Entertainment, Inc.* v. *Connectix Corp.*, 203 F.3d 596, 603 (9th Cir.), cert. denied, 531 U.S. 871 (2000). The Federal Circuit erred when it failed to draw a similar distinction here.

b. In holding that the declarations are copyrightable, the Federal Circuit also misapplied the merger doctrine.

That doctrine originated in this Court's seminal decision in *Baker* v. *Selden*, 101 U.S. (11 Otto) 99 (1880). The plaintiff had developed an accounting system and wrote a book explaining it. See *id.* at 100. His book included "certain forms or blanks, consisting of ruled lines, and headings, illustrating the system and showing how it is to be

used and carried out in practice." *Ibid.* The plaintiff contended that the forms were part of the book and therefore copyrightable. See *id.* at 101.

The Court rejected the plaintiff's argument. "The copyright of a work," the Court explained, "cannot give to the author an exclusive right to the methods of operation which he propounds, or to the diagrams which he employs to explain them." 101 U.S. at 103. A monopoly over those methods and diagrams could be secured only by patent law, not copyright, and in the absence of a patent, "any person may practise and use the art itself." *Id.* at 104. In short, "where the [useful] art [a work] teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public." *Id.* at 103.

In the context of computer programs, the merger doctrine "means that when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement." National Commission on New Technological Uses of Copyrighted Works, *Final Report of the National Commission on New Technological Uses of Copyrighted Works*, 3 Computer L.J. 53, 74 (1981). But the Federal Circuit took a completely different approach. In its view, the merger doctrine was inapplicable because "alternative expressions [we]re available" for the ideas embodied in the declarations. App., *infra*, 151a. Under that approach, however, the merger doctrine would be a nullity in the software context: it is difficult to see how any idea embodied in computer code could ever merge with its expression, because there will always be an alternative way of naming and stating the rules for

a specific function (such as determining the larger of two numbers).

Baker illustrates why that approach cannot be correct. It made no difference to the Court's analysis whether the defendant could have performed accounting generally without the plaintiff's forms, or whether the defendant could have developed his own, analogous accounting method. The critical point was that, having created an accounting system, the plaintiff disclosed it to the public. And because the plaintiff's forms were necessary for the public to use that precise system, they could not be copyrighted.

Here, as the district court found, using the Java API declarations was the only way to allow independent developers to rely on their preexisting knowledge of the Java language when creating new programs. App., *infra*, 103a-105a, 263a. If Google had not replicated the declarations exactly, developers' code that is "clearly [their] own work product" and was written using the industry-standard Java shorthand commands would not have run on Android. *Lotus*, 49 F.3d at 818. The developers would have been locked into the Java platform (which Oracle controlled) and would have been unable to reuse their own code, or their knowledge of familiar interfaces and commands, on the Android platform (or any other).

The First Circuit's decision in *Lotus* strongly supports the foregoing analysis. The spreadsheet program at issue allowed users to write customized programs, or "macros," that enabled them to execute a series of commands automatically by typing a single pre-programmed keystroke. See *Lotus*, 49 F.3d at 809. The defendant enabled prospective customers who had created their own macros in Lotus 1-2-3 to switch to its competing spreadsheet program without learning new commands or rewriting their macros. See *id.* at 810.

In *Lotus*, the First Circuit rejected the notion that Lotus could compel a user to "rewrite his or her macro using [another] program's menu command hierarchy." 49 F.3d at 818. As the court recognized, "forcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in § 102(b) that 'methods of operation' are not copyrightable." *Ibid.*; see *id.* at 819-820 (Boudin, J., concurring). So too here: the Federal Circuit's decision to extend copyright protection to the Java API declarations effectively grants Oracle a patent-like monopoly over the Java language.

In rejecting that reasoning, the Federal Circuit relied on the premise that Google could have given different names to Android's methods and libraries. App., *infra*, 153a-154a. Because Oracle had already selected names for Java's methods and libraries, however, the declarations could be written only in one way to permit Java-fluent developers to use the familiar shorthand commands. In any event, as the district court noted, it is well settled that such names and short phrases are not copyrightable. *Id.* at 264a.

The Federal Circuit compounded the error in its merger analysis by focusing exclusively on the choices available to Sun, Oracle's predecessor, at the time it created Java. With respect to the merger inquiry under Section 102(b), the question is "not whether *any* alternatives theoretically exist"; instead, it is "whether other options practically exist under the circumstances" and are "feasible within real-world constraints." *Lexmark*, 387 F.3d at 536.

The Federal Circuit should have taken account of the expressive choices available to *Google* when it created Android. Only then could it properly evaluate Google's claim that duplicating aspects of the Java API declarations was necessary to allow developers to create applications for

the Android platform using the free Java language, with the same functionality they were taught by Sun to expect of that language. See Pamela Samuelson, *Questioning Copyrights in Standards*, 48 B.C. L. Rev. 193, 215 (2007); *Lotus*, 49 F.3d at 819-820 (Boudin, J., concurring). For that additional reason, the Federal Circuit's merger analysis was deeply flawed, and its holding that the Java API declarations were copyrightable warrants further review.

B. This Court Should Grant Review To Decide Whether, As The Jury Found, Petitioner's Use Of A Software Interface In The Context Of Creating A New Computer Program Constitutes Fair Use

Even assuming that the Java API declarations were copyrightable, Google engaged in fair use when it used some of those declarations. In holding that Google's use was not fair as a matter of law, the Federal Circuit misapplied the precedents of this Court and others on the fair-use doctrine. In particular, the Federal Circuit failed to account for the functional nature of software interfaces; took an unduly constrained view of transformative use; and rendered the market-harm factor an essentially circular inquiry.

The Federal Circuit not only misapplied the fair-use doctrine; relying on its own (unsupported) findings, it also overturned a jury verdict along the way. In the earlier appeal in this case, the Federal Circuit remanded the fair-use issue to the jury out of "due respect for the limit of [its] appellate function." App., *infra*, 182a. After the retrial, the jury issued a verdict in Google's favor on fair use. At that point, the Federal Circuit did an about-face, taking the highly unusual step of setting aside the jury's verdict and deciding fair use in Oracle's favor as a matter of law. *Id.* at 53a-54a.

Because the jury returned a general verdict on fair use, the Federal Circuit correctly stated that it "must assume that the jury resolved all factual issues relating to the historical facts in favor of the verdict." App., *infra*, 23a. But the Federal Circuit said one thing and did another: it reconsidered for itself a number of factual issues presented to the jury and resolved those issues in support of the conclusion that Google's use was unfair as a matter of law. See p. 28, *infra*. To permit that approach would condone an unprecedented degree of appellate second-guessing of factual determinations in fair-use cases. This Court's intervention is urgently warranted to rectify the Federal Circuit's profoundly flawed approach.

1. The fair-use doctrine has long been a cornerstone of copyright law. That exception to copyright infringement, now codified in 17 U.S.C. 107, grew out of the recognition that new works "must necessarily borrow[] and use much which was well known and used before." *Campbell* v. *Acuff-Rose Music, Inc.*, 510 U.S. 569, 575 (1994) (citation omitted). As a result, "[f]rom the infancy of copyright protection, some opportunity for fair use of copyrighted materials has been thought necessary to fulfill copyright's very purpose." *Ibid.* To that end, the fair-use doctrine seeks to balance the "need simultaneously to protect copyrighted material and to allow others to build upon it." *Ibid.*

Four non-exclusive factors guide the fair-use analysis: (1) the purpose and character of the use; (2) the nature of the copyrighted work; (3) the substantiality of the portion used in relation to the copyrighted work as a whole; and (4) the effect on the potential market for or value of the copyrighted work. 17 U.S.C. 107. Those factors are to be "weighed together[] in light of the purposes of copyright." *Campbell*, 510 U.S. at 578. The ultimate aim in applying those factors is to "avoid rigid application of the copyright

statute" when "it would stifle the very creativity which that law is designed to foster." *Stewart* v. *Abend*, 495 U.S. 207, 236 (1990).

2. In its opinion on fair use, the Federal Circuit engaged in precisely the kind of rigid application that this Court warned against. It made at least three critical errors, one that infected the entire analysis and two that pertained to the fair-use factors that drove its reasoning—the purpose and character of the use and the effect on the actual or potential market.

a. As a matter of overall approach, the Federal Circuit failed to adapt the fair-use doctrine to the functional nature of software interfaces. Each software interface is designed to trigger the performance of a specific operation, such as finding the larger of two numbers. In considering the second fair-use factor, the nature of the copyrighted work, the Federal Circuit acknowledged that "functional considerations were both substantial and important" and thus that the second factor "favors a finding of fair use." App., *infra*, 42a. But the Federal Circuit ignored how the functional nature of software interfaces affects the fair-use analysis as a whole.

A functional work is entitled only to thin copyright protection—which, in turn, makes it easier to prove a noninfringing, fair use of that work. This Court has recognized that "some works are closer to the core of intended copyright protection than others, with the consequence that fair use is more difficult to establish when the former works are copied." *Campbell*, 510 U.S. at 586. Unlike literary or artistic works, software interfaces "perform[] functions that are not entitled to copyright protection." *Sony*, 203 F.3d at 602. In light of that functional character, interfaces lie "at a distance from the core" of copy-

right protection and are thus owed a "lower degree of protection than more traditional literary works." *Id.* at 603; see *Sega*, 977 F.2d at 1527.

In its opinion on fair use, the Federal Circuit seemingly took a contrary position. While recognizing the functional nature of software interfaces, the Federal Circuit gave them the same copyright protection—and, as is relevant here, the same fair-use treatment—afforded to literary and artistic works. The Federal Circuit thus systematically erred when it discounted the particular characteristics of software interfaces.

b. Beyond that overarching flaw, the Federal Circuit applied the fair-use factors incorrectly and too rigidly. That is most apparent in its consideration of the first fair-use factor, the purpose and character of the use—and, in particular, a component of that factor, transformative use. Transformative use focuses on "whether the new work merely supersede[s] the objects of the original creation, or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message." *Campbell*, 510 U.S. at 579 (internal quotation marks and citations omitted).

The Federal Circuit's most fundamental mistake in analyzing transformative use was in fixating only on the material that Google reused (certain Java API declarations) and assessing whether that material was itself transformed in the new work. App., *infra*, 32a-37a. That is not the correct inquiry. As numerous courts have indicated, "a secondary work can be transformative in function or purpose without altering or actually adding to the original work." *American Society for Testing & Materials* v. *Public.Resource.Org, Inc.*, 896 F.3d 437, 450 (D.C. Cir. 2018); see *Swatch Group Management Services Ltd.* v. *Bloomberg L.P.*, 756 F.3d 73, 84 (2d Cir. 2014); *Seltzer* v. *Green Day*, 725 F.3d 1170, 1177 (9th Cir. 2013).

Instead, the correct inquiry focuses on the "new work" *as a whole*, asking whether the "new work * * * adds something new [to the copyrighted work], with a further purpose or different character." *Campbell*, 510 U.S. at 579. That new purpose or character, in turn, informs the degree and nature of the transformation necessary to qualify as fair use: for example, a parody "needs to mimic an original to make its point," which may include copying the heart of the original work to "make the object of its critical wit recognizable." *Id.* at 580-581, 588.

Here, both the new work Google created and its use of the Java API declarations were undoubtedly transformative. Google set out to create an entirely new platform for smartphones. In contrast, the original work, Java SE, was designed for desktop and server computers. The new Android platform would have to accommodate resource constraints, such as limited memory and battery life, that did not apply to Java SE. See p. 4, *supra*. The Federal Circuit therefore erred when it suggested that Google's new platform merely changed existing computer code from one medium or format to another. App., *infra*, 35a-37a.

The new work also used the Java API declarations in a transformative way. The declarations formed only a small part of the new work. Because the Android platform needed to be tailored to a new smartphone environment, Google had to develop all of its own implementing code for Android, including new implementing code for the Java API declarations. App., *infra*, 112a, 114a. Google created entirely new libraries, including new declarations and implementing code, for functions necessary to operate modern smartphones, such as touchscreens, web browsing, the built-in camera, and location awareness.

To be sure, Google did incorporate certain Java API declarations from Java SE. That limited incorporation allowed developers to use the Java language to build applications for Android. By the same token, Google's use of the declarations prevented Oracle from locking in developers familiar with the Java language into building applications only for Oracle's platforms. Notably, the government recognized those "legitimate concerns" of lock-in effects and interoperability in addressing Google's earlier petition for certiorari in this case, and it highlighted fair use as the proper way to accommodate those concerns. 14-410 U.S. Br. 17.

Such interoperability was critical for developers programming in the Java language. At Sun's and Oracle's encouragement, developers had invested in learning the Java language and had grown accustomed to using the well-known shorthand commands derived from the Java API declarations. The district court likened those declarations to the keys on a QWERTY keyboard. App., *infra*, 104a. Developers therefore wanted to use the Java API declarations to write code for Android applications in the Java language. To allow such code to run on Android, Google had to incorporate the applicable Java API declarations. By allowing applications written in the Java language to operate in the new environment, those declarations took on a "further purpose or different character" in Android that they did not have in Java SE. *Campbell*, 510 U.S. at 579.

In the Federal Circuit's view, Google's incorporation of the software interfaces into a new work was effectively irrelevant for fair-use purposes, because the Java API declarations performed the same function in the original and new works. App., *infra*, 33a. That is a dangerous misapplication of the fair-use doctrine with breathtakingly broad implications. If a mere identity of function were

enough to preclude fair use, the reuse of any preexisting computer code in new software would never fall within the fair-use defense. That is because computer code is essentially a set of instructions that performs the same function whenever it is used. See, *e.g.*, *Sega*, 977 F.2d at 1524.

Here, the Java API declarations operated via entirely new implementing code in Android. For instance, the declaration may call for something to be displayed, and the corresponding implementing code would display the output on the touchscreen of a smartphone, rather than the monitor of a desktop computer. The reuse of the Java API declarations in Android simply permitted developers to program Android applications in the free and open Java language.

The Federal Circuit effectively dismissed the concerns regarding lock-in effects and interoperability that the government has recognized as "substantial and important," 14-410 U.S. Br. 17, and that other courts have similarly emphasized, see, *e.g.*, *Sega*, 977 F.2d at 1526-1527. If Google had used entirely different interfaces in Android, developers would have had to learn the new interfaces to operate the prewritten methods that they already knew from the Java language. Developers could not have used the familiar, industry-standard Java shorthand commands to build Android applications and would therefore have been deterred from doing so.

As a practical matter, then, precluding Google's use of the Java API declarations would permit Oracle to accrue market power via copyright, locking in developers that had invested in learning the Java language and making it difficult for them to use those skills to program for new platforms. See *Lotus*, 49 F.3d at 821 (Boudin, J., concurring). Oracle's control, in turn, would effectively block competing platforms from accessing developers trained in the Java language. Because interfaces are central to how

developers operate software, control over interfaces gives rise to barriers to entry and implicates issues of competition and innovation that warrant this Court's review.

c. The Federal Circuit's approach to the fourth fair-use factor—the effect on the actual or potential market for the copyrighted work—was similarly flawed. As this Court has explained, the fair-use doctrine applies to "copying by others which does not materially impair the marketability of the work which is copied." *Harper & Row Publishers, Inc.* v. *Nation Enterprises*, 471 U.S. 539, 566-567 (1985). The Federal Circuit concluded that Google's use of the Java API declarations caused harm to Oracle's *actual* market because Java SE was purportedly also used in mobile phones before Android's debut. App., *infra*, 50a-51a. And even if there was no actual harm, the Federal Circuit concluded that there was potential harm because smartphones were a "traditional, reasonable, or likely to be developed market" for Java SE. *Id.* at 51a-52a.

The Federal Circuit reached those conclusions only by improperly revisiting and reversing the jury's implicit factual determinations. Whether and to what extent Java SE was used in mobile devices were disputed issues below. The Federal Circuit acknowledged that it "must assume that the jury resolved all factual issues relating to the historical facts in favor of the verdict." App., *infra*, 23a. Yet it rejected that assumption and instead made its own (erroneous) determination that Java SE was in fact used in early mobile devices comparable to Android before Android's release. See *id.* at 50a-51a. If the Federal Circuit had deferred to the jury, as it was required to do, it would have concluded that Java SE was never used in a modern smartphone and that Java SE and Android occupied different markets, which meant that there was no actual market harm.

The Federal Circuit further concluded that, simply because Oracle *could* have tried to adapt Java SE for use in smartphones, Google's use of the Java API declarations in a mobile platform caused harm to a potential market. App., *infra*, 51a-52a. To find market harm on that basis, however, is entirely circular. The fair-use issue arose in this case precisely because Google did not "pay a fee for the right to * * * use" the declarations. *Bill Graham Archives* v. *Dorling Kindersley Ltd.*, 448 F.3d 605, 614 (2d Cir. 2006). If the potential market for the copyrighted material is defined to include the market for licensing the exact use at issue, then potential market harm is baked into the market definition. Courts and commentators alike have warned against watering down the market-harm inquiry in that way. See, *e.g.*, 4 Nimmer § 13.05[A][4]; *Swatch Group*, 756 F.3d at 91.

In sum, the Federal Circuit's fair-use analysis is replete with errors and cannot be reconciled with the decisions of this Court and others. Further review is warranted on the fair-use question, as well as the copyrightability question.

C. The Questions Presented Are Exceptionally Important And Warrant Review In This Case

1. Above and beyond the broader implications for copyright law, this case warrants the Court's attention for its sheer practical importance. Repeatedly hailed as the "copyright lawsuit of the decade," this case presents a conflict between two giants of the technology industry, Google and Oracle. At the center of this dispute is Android, a platform used worldwide by billions of users. And what Oracle is seeking here is nothing less than complete control over a community of developers that have invested in learning the free and open Java language. That effort

aims to stifle rather than encourage the creation of new works.

Given the enormous stakes, it is unsurprising that this case has drawn widespread attention, including from the Court. The last time that Google sought review in this case—when the sole question involved copyrightability—this Court called for the views of the Solicitor General. While the government advised against granting certiorari while the case was in an interlocutory posture, it noted "substantial and important concerns" that in its view should be addressed through the fair-use doctrine. 14-410 U.S. Br. 17. Numerous industry groups, academics, and other interested parties filed amicus briefs in the proceedings below. And numerous commentators have highlighted the widespread impact of the Federal Circuit's decision and stressed the need for this Court's intervention. See, *e.g.*, Tony Dutra, *Oracle Victory in Copyright Case Has Seeds for a Google Appeal*, Bloomberg Law (Mar. 28, 2018) <tinyurl.com/dutraarticle>.

As those amici and commentators have warned, if allowed to stand, the Federal Circuit's approach would have a devastating impact on the development of computer software. Although this case involves software interfaces, the Federal Circuit's reasoning has implications for all computer code. In particular, if code must perform a different function in a new work for the fair-use defense to apply, then a developer will be foreclosed from reusing copyrighted code designed to execute one particular function.

The Federal Circuit's reasoning threatens the prevailing approach to building computer software. Developers are not coding programs entirely from scratch, as they may have been in the early days of programming. Instead, new programs now incorporate and rely on preexisting interfaces to trigger certain functions, which saves

the wasted effort of reinventing and retesting what came before. See David Orenstein, *Application Programming Interface*, ComputerWorld (Jan. 10, 2000) <tinyurl.com/orensteinarticle>. If the Federal Circuit's approach is allowed to stand, developers will be forced to abandon their traditional building-block approach to software development. At the very least, they will be left in confusion about whether and when their longstanding practices constitute copyright infringement.

Not only does the Federal Circuit's approach wreak havoc on copyright law, but it also risks disturbing the balance between copyright law and patent law, the two principal bodies of law that govern innovation. The Federal Circuit has effectively provided blanket copyright protection to an entire class of computer code. It has done so despite repeated warnings from other courts that patent law may be better equipped to address the functional aspects of computer code. See, *e.g.*, *Sony*, 203 F.3d at 605; *Computer Associates*, 982 F.2d at 712; *Sega*, 977 F.2d at 1526. As those courts have pointed out, if the creator of computer code "wishes to obtain a lawful monopoly on the *functional concepts* in its software, it must satisfy the more stringent standards of the patent laws," including novelty and nonobviousness. *Sony*, 203 F.3d at 605 (emphasis added). In contrast, "copyright registration—with its indiscriminating availability—is not ideally suited to deal with the highly dynamic technology of computer science." *Computer Associates*, 982 F.2d at 712.

The Federal Circuit's approach pays no heed to those warnings. In the opinions under review, the Federal Circuit afforded software interfaces a government-granted monopoly based on a more relaxed standard and for a much longer period than permitted by patent law. Software interfaces—the critical building blocks of software development—can now be kept out of the public domain

for at least 70 years after the creator's death. This Court should closely scrutinize the Federal Circuit's expansion of copyright law into the traditional territory of the patent system.

2. This case is an ideal vehicle for considering the questions presented. The Federal Circuit has now squarely held both that software interfaces are copyrightable and that petitioner's reuse of such interfaces did not constitute fair use as a matter of law. The questions presented have been exhaustively briefed by the parties and their amici below.

There would be no material benefit from further percolation on the questions presented. As to copyrightability, there has been a persistent circuit conflict, and the arguments for both sides have been amply considered by the courts of appeals. See pp. 12-17, *supra*. And as to fair use, the numerous errors in various aspects of the Federal Circuit's analysis warrant the Court's intervention. That is particularly true because a copyright holder will be able to invoke the Federal Circuit's jurisdiction through the simple expedient of including a patent claim. See, *e.g.*, Joe Mullin, *Cisco v. Arista Awaits a Jury Verdict Under the Oracle v. Google Shadow*, ArsTechnica (Dec. 14, 2016) <tinyurl.com/y9xxd4zf> (noting the use of the same strategy in another case).

In short, the Federal Circuit has deepened a widely recognized conflict on the copyrightability of software interfaces and effectively excluded their reuse from the fair-use defense. The courts of appeals have taken divergent approaches to the application of copyright law to computer software, a key driver of technological innovation. The Federal Circuit should not have the final word in this landmark case. This Court's review is unquestionably warranted.

CONCLUSION

The petition for a writ of certiorari should be granted.

Respectfully submitted.

THOMAS C. GOLDSTEIN
GOLDSTEIN & RUSSELL, P.C.
5225 Wisconsin Avenue, N.W., Suite 404
Washington, DC 20015

ROBERT A. VAN NEST
CHRISTA M. ANDERSON
EUGENE M. PAIGE
REID P. MULLEN
KEKER, VAN NEST & PETERS LLP
633 Battery Street
San Francisco, CA 94111

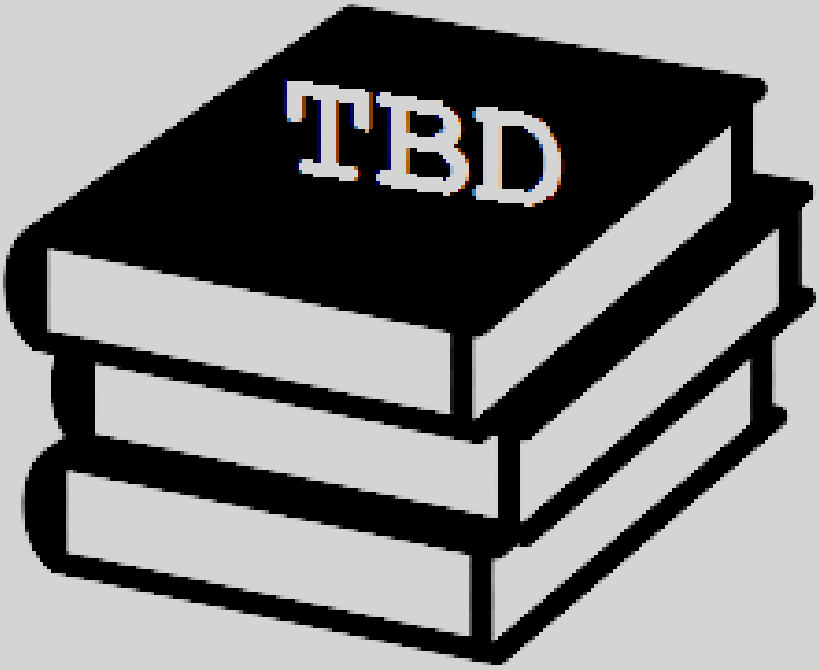
BRUCE W. BABER
MARISA C. MALECK
KING & SPALDING LLP
1180 Peachtree Street, N.E.
Atlanta, GA 30309

KANNON K. SHANMUGAM
CHARLES L. MCCLOUD
MENG JIA YANG
WILLIAMS & CONNOLLY LLP
725 Twelfth Street, N.W.
Washington, DC 20005
(202) 434-5000
kshanmugam@wc.com

MICHAEL S. KWUN
KWUN BHANSALI LAZARUS LLP
555 Montgomery Street, Suite 750
San Francisco, CA 94111

JANUARY 2019

BUY™	SELL™	SHOP™



SELL YOUR OWN SAMPLES

(help others get the justice that they deserve)

BUY™	SELL™	SHOP™