

No. 14-____

In the
Supreme Court of the United States

GOOGLE INC.,

PETITIONER,

v.

ORACLE AMERICA, INC.,

RESPONDENT.

On Petition for a Writ of Certiorari
to the United States Court of Appeals
for the Federal Circuit

PETITION FOR A WRIT OF CERTIORARI

BRUCE W. BABER
KING & SPALDING LLP
1180 Peachtree Street, NE
Atlanta, GA 30309

ROBERT A. VAN NEST
STEVEN A. HIRSCH
CHRISTA M. ANDERSON
MICHAEL S. KWUN
DAN JACKSON
KEKER & VAN NEST LLP
633 Battery Street
San Francisco, CA 94111

DARYL L. JOSEFFER
Counsel of Record
ASHLEY C. PARRISH
ADAM M. CONRAD
ETHAN P. DAVIS
KING & SPALDING LLP
1700 Pennsylvania Ave., NW
Washington, DC 20006
(202) 737-0500
djoseffer@kslaw.com

RENNY HWANG
GOOGLE INC.
1600 Amphitheatre Parkway
Mountain View, CA 94043

Counsel for Petitioner

October 6, 2014

QUESTION PRESENTED

Congress specified that “original works of authorship” are generally eligible for copyright protection, 17 U.S.C. § 102(a), but “[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.” *Id.* § 102(b).

In this case, the Federal Circuit held that Section 102(b) does *not* exclude systems or methods of operation from copyright protection and that all elements of an original work are “entitled to copyright protection as long as the author had multiple ways to express the underlying idea.” App. 47.

The question is:

Whether copyright protection extends to all elements of an original work of computer software, including a system or method of operation, that an author could have written in more than one way.

**PARTIES TO THE PROCEEDING
AND RULE 29.6 STATEMENT**

Petitioner in this Court, defendant-cross appellant below, is Google Inc. Respondent in this Court, plaintiff-appellant below, is Oracle America, Inc.

Google Inc. is a publicly traded company (NASDAQ: GOOG and GOOGL). No publicly held company owns 10 percent or more of Google Inc.'s stock.

TABLE OF CONTENTS

| | |
|--|-----|
| QUESTION PRESENTED..... | i |
| PARTIES TO THE PROCEEDING AND RULE 29.6 STATEMENT..... | ii |
| TABLE OF AUTHORITIES..... | vii |
| PETITION FOR A WRIT OF CERTIORARI..... | 1 |
| OPINIONS BELOW | 4 |
| JURISDICTION | 4 |
| STATUTORY PROVISION INVOLVED | 4 |
| STATEMENT OF THE CASE | 5 |
| A. Java and Android | 5 |
| B. The District Court Proceedings | 8 |
| C. The Court of Appeals Proceedings..... | 10 |
| REASONS FOR GRANTING THE PETITION..... | 13 |
| I. The Courts Of Appeals Are In Disarray About The Application Of Section 102(b) To Software..... | 13 |
| II. The Federal Circuit’s Decision Runs Afoul Of The Statute, This Court’s Controlling Precedents, And The Distinction Between Patent And Copyright. | 20 |
| A. The statute codifies this Court’s exclusion of systems and methods of operation from copyright protection. | 20 |
| B. Systems and methods of operation are governed by patent, not copyright, law. | 23 |

| | |
|--|--------|
| C. The Java method headers are a system or method of operation. | 29 |
| III. This Case Presents a Recurring Question of Exceptional Importance. | 32 |
| CONCLUSION | 37 |
| APPENDIX | |
| Appendix A | |
| Opinion of the United States Court of Appeals for the Federal Circuit (May 9, 2014) | App-1 |
| Appendix B | |
| Order Partially Granting and Partially Denying Defendant’s Motion for Summary Judgment on Copyright Claim of the United States District Court for the Northern District of California (Sept. 15, 2011)..... | App-79 |
| Appendix C | |
| Order on Motions for Judgment as a Matter of Law of the United States District Court for the Northern District of California (May 10, 2012) | App-99 |

Appendix D

Order Regarding Copyrightability of
Certain Replicated Elements of the
Java Application Programming
Interface of the United States
District Court for the Northern
District of California (May 31, 2012)..... App-100

Appendix E

Findings of Fact and Conclusions of
Law on Equitable Defenses of the
United States District Court for the
Northern District of California (May
31, 2012)..... App-165

Appendix F

Final Judgment of the United States
District Court for the Northern
District of California (June 20, 2012) App-169

Appendix G

Order Denying Motion for Judgment
as a Matter of Law and New Trial of
the United States District Court for
the Northern District of California
(July 13, 2012) App-172

Appendix H

Order Denying Motion for Judgment
as a Matter of Law and New Trial of
the United States District Court for
the Northern District of California
(Sept. 4, 2012) App-173

Appendix I

| | |
|-----------------------|---------|
| 17 U.S.C. § 101 | App-174 |
| 17 U.S.C. § 102 | App-188 |

TABLE OF AUTHORITIES

Cases

| | |
|---|----------------|
| <i>Alice Corp. Pty. Ltd. v. CLS Bank Int’l</i> , 134 S. Ct. 2347 (2014) | 25, 27, 28 |
| <i>Am. Broad. Cos. v. Aereo, Inc.</i> , 134 S. Ct. 2498 (2014) | 37 |
| <i>Apple Computer, Inc. v.</i> <i>Franklin Computer Corp.</i> , 714 F.2d 1240 (3d Cir. 1983)..... | 17, 18 |
| <i>Ass’n for Molecular Pathology v.</i> <i>Myriad Genetics, Inc.</i> , 133 S. Ct. 2107 (2013) | 28 |
| <i>ATC Distribution Grp., Inc. v. Whatever It Takes</i> <i>Transmissions & Parts, Inc.</i> , 402 F.3d 700 (6th Cir. 2005) | 16 |
| <i>Baker v. Selden</i> , 101 (11 Otto) U.S. 99 (1880)..... | passim |
| <i>Bilski v. Kappos</i> , 130 S. Ct. 3218 (2010) | 28 |
| <i>Brief English Sys., Inc. v. Owen</i> , 48 F.2d 555 (2d Cir. 1931)..... | 29, 30 |
| <i>Computer Assocs. Int’l Inc. v. Altai, Inc.</i> , 982 F.2d 693 (2d Cir. 1992)..... | 14, 17, 18 |
| <i>Eng’g Dynamics, Inc. v. Structural Software, Inc.</i> , 26 F.3d 1335 (5th Cir. 1994) | 17 |
| <i>Eng’g Dynamics, Inc. v. Structural Software, Inc.</i> , 46 F.3d 408 (5th Cir. 1995) | 17 |
| <i>Feist Publ’ns, Inc. v. Rural Tel. Serv. Co.</i> , 499 U.S. 340 (1991) | 10, 21, 22, 26 |

| | |
|---|------------|
| <i>Gates Rubber Co. v. Bando Chem. Indus.</i> , 9 F.3d 823 (10th Cir. 1993) | 17 |
| <i>Kregos v. Associated Press</i> , 937 F.2d 700 (2d Cir. 1991)..... | 19 |
| <i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007) | 25 |
| <i>Lexmark Int’l, Inc. v.</i> <i>Static Control Components, Inc.</i> , 387 F.3d 522 (6th Cir. 2004) | 15, 16, 19 |
| <i>Lotus Dev. Corp. v. Borland Int’l, Inc.</i> , 49 F.3d 807 (1st Cir. 1995)..... | passim |
| <i>Lotus Dev. Corp. v. Borland Int’l, Inc.</i> , 516 U.S. 233 (1996) | 1 |
| <i>Lotus Dev. Corp. v. Borland Int’l, Inc.</i> , 831 F. Supp. 223 (D. Mass. 1993) | 29 |
| <i>Matthew Bender & Co. v. West Publ’g Co.</i> , 158 F.3d 674 (2d Cir. 1998)..... | 19 |
| <i>Mayo Collaborative Servs. v.</i> <i>Prometheus Labs., Inc.</i> , 132 S. Ct. 1289 (2012) | 25, 28 |
| <i>Milner v. Dep’t of Navy</i> , 131 S. Ct. 1259 (2011) | 22 |
| <i>Mitel, Inc. v. Iqtel, Inc.</i> , 124 F.3d 1366 (10th Cir. 1997) | 14, 18 |
| <i>North Coast Indus. v. Jason Maxwell, Inc.</i> , 972 F.2d 1031 (9th Cir. 1992) | 19 |
| <i>Publications Int’l, Ltd. v. Meredith Corp.</i> , 88 F.3d 473 (7th Cir. 1996) | 19 |

| | |
|---|--------|
| <i>Research Corp. Techs., Inc. v. Microsoft Corp.</i> , 627 F.3d 859 (Fed. Cir. 2010)..... | 27 |
| <i>SAS Institute Inc. v. World Programming Ltd.</i> Case C-406/10, 2012 E.C.L.I 259, [2012] 3 C.M.L.R. 4..... | 36 |
| <i>Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.</i> , 797 F.2d 1222 (3d Cir. 1986)..... | 14, 16 |
| <i>Yankee Candle Co. v. Bridgewater Candle Co.</i> , 259 F.3d 25 (1st Cir. 2001)..... | 19 |
| Statutes | |
| 17 U.S.C. § 101 | 27 |
| 17 U.S.C. § 102 | passim |
| 17 U.S.C. § 107 | 28 |
| 17 U.S.C. § 302 | 26 |
| 17 U.S.C. § 1201 | 35 |
| 28 U.S.C. § 1254 | 4 |
| 35 U.S.C. § 101 | 25 |
| 35 U.S.C. § 102 | 25 |
| 35 U.S.C. § 103 | 25 |
| Other Authorities | |
| Amazon Web Services, <i>AWS Products & Solutions</i> (2014) | 34 |
| Br. of <i>Amici Curiae</i> Rackspace US, Inc. <i>et al.</i> , <i>Oracle Am., Inc. v. Google Inc.</i> , Dkt. No. 116, Nos. 13-1021, <i>et al.</i> (Fed. Cir. May 30, 2013)..... | 34 |

| | |
|--|----|
| Council of Ministers Directive 91/250/EEC of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J. (L 122) | 35 |
| Ferguson, Charles H. & Morris, Charles R., <i>Computer Wars: The Post-IBM World</i> (1994).... | 33 |
| H.R. REP. NO. 94-1476 (1976)..... | 22 |
| International Data Corporation, Worldwide Smartphone OS Market Share (2014) | 7 |
| Kane, Tim, Ewing Marion Kauffman Foundation, <i>The Importance of Startups in Job Creation and Job Destruction</i> (2010) | 36 |
| Lindberg, Van, <i>The Copyrightability Of APIs In The Land Of OpenStack</i> (2014)..... | 33 |
| Oracle Corp., <i>Frequently Asked Questions, Oracle Linux</i> (2014) | 33 |
| S. REP. NO. 94-473 (1975)..... | 22 |
| Vaughan-Nichols, Steven J., <i>OpenStack vs. CloudStack: The Beginning of the Open-Source Cloud Wars</i> , ZDNet (Apr. 12, 2012) | 34 |
| Wilcox, Joe, <i>Will OS X's Unix Roots Help Apple Grow?</i> , CNET.com (May 21, 2001) | 33 |
| WineHQ, <i>About Wine</i> | 35 |

PETITION FOR A WRIT OF CERTIORARI

In 1995, this Court granted certiorari in *Lotus Development Corp. v. Borland International, Inc.*, 516 U.S. 233 (1996), to resolve the question presented here. The First Circuit had held—consistent with the plain language of 17 U.S.C. § 102(b) but in conflict with other courts of appeals—that methods of operation embodied in computer programs are not entitled to copyright protection. This Court deadlocked, affirming by an equally divided court. Two decades later, this oft-acknowledged circuit split has deepened and the question presented has grown even more important as software has become a fixture of modern life.

This case directly implicates the unanswered question in *Lotus* because the Federal Circuit extended copyright protection to systems and methods of operation, including computer interfaces. That holding would obstruct an enormous amount of innovation in fast-moving, high-technology industries, in part because innovation depends on software developers' ability to build on what has come before. If the Federal Circuit's holding had been the law at the inception of the Internet age, early computer companies could have blocked vast amounts of technological development by claiming 95-year copyright monopolies over the basic building blocks of computer design and programming. By the time Google and countless other innovators even came onto the scene, others could have locked up the field for longer than most people will live.

Consider, for example, the well-known keyboard design known as QWERTY. After Remington

developed that organization of letters and symbols decades ago, it became standard for typewriters and, later, for computer keyboards. People invested time and effort in learning the QWERTY design, and then expected all keyboards to use it. Later, companies like IBM and Apple added their own additional keys to the original QWERTY layout. If Remington had brought a copyright infringement lawsuit against a keyboard manufacturer for copying the QWERTY layout, it would have failed. That design was original and creative, but Remington was not entitled to appropriate the investments made by others in learning how to use it. Otherwise, Remington could have monopolized not only the sale of its patented typewriters for the length of a patent term, but also the sale of all keyboards for nearly a century.

This case raises the same basic issue. Individual computer programmers and third-party companies develop applications (the ubiquitous “apps”) for mobile devices, such as smartphones, that use the Android platform. Because many computer programmers are familiar with the Java programming language, Google allowed programmers to write programs for Android using it, including the basic shorthand commands of the Java language. As relevant here, a person writing an Android application in the Java language may use shorthand commands to cause a computer to perform certain functions, such as choosing the larger of two numbers. Programmers have made significant investments in learning these commands; they are, in effect, the basic vocabulary words of the Java language. When programmers sit down to write applications, they expect to be able to use them.

The Federal Circuit nonetheless held that, although the Java language is concededly not entitled to copyright protection, the elements of the Java platform that enable the use of the shorthand commands are copyrightable. The court based that conclusion on its view that 17 U.S.C. § 102(b) does *not* exclude systems and methods of operation from copyright protection—even though the statute unambiguously does exactly that:

In 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).

By replacing that statutory directive with a different one—that copyright protection *does* extend to a system or method of operation so long as there was more than one way to write it—the Federal Circuit usurped Congress’s role, deepened a circuit split that this Court previously granted certiorari to resolve, allowed Oracle to use copyright law to evade the limits on patent protection, and thereby blocked developers from building on what has come before. The court did so, moreover, in one of the most important cases of its kind, concerning the widely-used Java language and Android platform. This Court’s review is needed now, before tomorrow’s innovation falls victim to the decision below.

OPINIONS BELOW

The opinion of the court of appeals is reported at 750 F.3d 1339 and reproduced at App. 1. The district court's opinion is published at 872 F. Supp. 2d 974 and reproduced at App. 100.

JURISDICTION

The court of appeals rendered its decision on May 8, 2014. On July 10, 2014, the Chief Justice extended the time for filing a petition to and including October 6, 2014. This Court has jurisdiction under 28 U.S.C. § 1254(1).

STATUTORY PROVISION INVOLVED

Section 102 of the Copyright Act provides:

(a) Copyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. . . .

. . . .

(b) In 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.

STATEMENT OF THE CASE

A. Java and Android

1. Sun Microsystems released the Java programming language and software platform in 1996. By making the Java language free for all to use, Sun sought to “build the biggest tent and invite as many people as possible.” C.A. App. 22141.

As the district court explained, the Java language is made up of “keywords and other symbols” as well as “a set of pre-written programs to carry out various commands.” App. 106. In encouraging computer programmers to learn and use Java, Sun touted those pre-written programs. C.A. App. 22137. Sun succeeded in bringing an entire generation of programmers into the Java community. App. 105. Millions of programmers invested time and effort into learning Java, making it one of the world’s most popular programming languages. App. 104.

Programmers access the set of pre-written programs through the Java application programming interface (“API”)—a highly structured system with its own nomenclature. The application programming interface provides access to thousands of “methods,” each of which performs a function such as choosing the higher of two numbers. The methods are grouped into “classes,” which are further grouped into 166 “packages” of programs—much like members of the animal kingdom are grouped into species, genres, and families. *See* App. 106–07.

The computer code for each method “consists of the method header and the method body.” App 111. The method header, also known as a “declaration,”

“introduces the method body” and “specif[ies] the names, parameters and functionality of the methods and classes.” App. 7, 29–30. “The method body is a block of code that then implements the method” by instructing a computer how to perform the relevant function; it is therefore known as “implementing code.” App. 111.

To use the methods, programmers do not need to concern themselves with the methods’ implementing code. Instead, programmers use a shorthand command that causes the implementing code to perform the desired function, such as choosing the greater of two numbers. App. 33. In this way, a programmer uses the shorthand commands to operate the methods, *i.e.*, the pre-written programs. By using a method’s shorthand command, a programmer can write complex software efficiently, without having to write out implementing code for each individual routine task.

These shorthand commands take the specific format “java.package.Class.method(input).” App. 112–16. For example, “java.lang.Math.max(1,2)” refers to a particular method (“max”) that returns the greater of two numbers (*i.e.*, 1 and 2) and is located in the “Math” class, which in turn is located in the “java.lang” package. App. 112. Each shorthand command is derived from the method’s header, which, like the command, specifies the method’s name, class, package, and inputs. App. 7, 29–30.

2. Google is the lead developer of Android, one of the most popular mobile device platforms in the world. In the second quarter of 2014, third-party manufacturers such as Samsung, HTC, LG, and

Lenovo sold more than 255 million smartphones that use the Android platform. *See* International Data Corporation, *Worldwide Smartphone OS Market Share* (2014), *available at* <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.

The Android platform includes 168 packages of methods. App. 109. For every one of those methods, Google wrote or acquired original implementing code. App. 101. As the district court explained, “[a]ll agree that Google was and remains free to use the Java language itself” and that the “method implementations by Google are free of copyright issues.” App. 108. The parties’ dispute centers on Google’s use of the same headers for the methods found in 37 of the Android packages—methods that perform “functions . . . that [a]re key to mobile devices.” App. 107.

Independent computer programmers create applications for use on Android devices. Because those programmers know and often prefer to use the Java programming language, Google concluded that programmers “would want to find the same 37 sets of functionalities in the new Android system callable by the same names as used in Java.” App. 9. For those shorthand commands to work on the Android platform, Google had to replicate the method headers precisely; any change to the headers would have prevented the shorthand commands from working properly. App. 109–10. As the district court found, therefore, “Android and Java *must be* identical when it comes to those particular lines of code.” App. 109. Because Google replicated only the method headers, and the body of each method (the implementing code)

was written from scratch, “only three percent of the lines of code are the same” in the 37 disputed packages. App. 109.

B. The District Court Proceedings

After Oracle acquired Sun in 2010, Oracle brought this action for patent and copyright infringement. The district court entered judgment in Google’s favor on Oracle’s patent claims, and Oracle has not appealed that determination. App. 170.

Oracle’s copyright claims accused Google of copying the method headers and the so-called “structure, sequence, and organization” of the Java application programming interface. App. 3. Oracle premised its “structure, sequence, and organization” claim on the theory that the method headers “embody the structure” of the application programming interface by specifying the name, package, and class of each method. App. 21. All of Oracle’s claims thus challenged the same thing: Google’s replication of the method headers. App. 101. Google responded, in part, that Java’s method headers are not entitled to copyright protection because, among other things, they constitute or embody a system or method of operation—specifically, a system or method of operating the pre-written programs.

The district court considered the copyrightability of the method headers at the same time the jury considered whether—*if* the district court held the method headers to be copyrightable—Google would be liable for infringement. Those two determinations proceeded on parallel tracks, with the district court instructing the jury to assume that Oracle was

entitled to copyright protection and to consider only infringement and fair use. The jury found in Oracle's favor on infringement but hung on Google's fair-use defense. App. 12.

In an extensive published opinion, the district court held that the method headers are not copyrightable and that Google is therefore entitled to judgment as a matter of law. App. 100–65. The court emphasized that Google was entitled to write its own code implementing the same functions or methods that are found in the Java application programming interface. “[C]opyright law does not confer ownership over any and all ways to implement a function or specification, no matter how creative [it] may be.” App. 154.

The district court then held that the method headers, including their names and organization, are a system or method of operation excluded from copyright protection under Section 102(b) of the Copyright Act. App. 159. Because the system of method headers is a “command structure” for operating the pre-written programs, the court concluded that it might receive “patent protection perhaps—but not copyright protection.” *Id.*

The district court emphasized that compatibility “sheds further light on the character of the command structure as a system or method of operation.” App. 159. By the time Android came into existence, programmers had written “millions of lines of code” in Java, which “necessarily used the `java.package.Class.method()` command format” and “called on all or some of the specific 37 packages at issue and necessarily used the command structure of names

[used by Google].” *Id.* “In order for at least some of this code to run on Android, Google was required to [use] the same `java.package.Class.method()` command system using the same names with the same ‘taxonomy’ and with the same functional specifications.” App. 159–60. As a result, “Google replicated what was necessary to achieve a degree of interoperability—but no more.” App. 160.

The district court found further support for its holding in other principles of copyright law. First, “[u]nder the merger doctrine, when there is only one (or only a few) ways to express something, then no one can claim ownership of such expression by copyright.” App. 153. Second, “names and short phrases are not copyrightable.” *Id.* Third, citing this Court’s decision in *Feist Publications, Inc. v. Rural Telephone Service Co.*, 499 U.S. 340, 356 (1991), the court observed that “we should not yield to the temptation to find copyrightability merely to reward an investment made in a body of intellectual property.” App. 153.

C. The Court of Appeals Proceedings

The Federal Circuit reversed, opining that copyrightability presents “a low bar” that requires only that a work be original and expressive in the sense that “the author had multiple ways to express the underlying idea.” App. 17, 47. The court noted a three-way circuit split on whether to deny copyright protection to all systems or methods of operation, grant copyright protection to essentially all elements of an original and creative computer program (including systems and methods of operation), or

apply a third test known as the abstraction/filtration/comparison test. App. 23–24.

Applying Ninth Circuit law because this case arose within that circuit and copyright law does not fall within the Federal Circuit’s exclusive jurisdiction, the Federal Circuit concluded that the Ninth Circuit has adopted the abstraction/filtration/comparison test. App. 24. After identifying a circuit split on how to apply that test, the court of appeals explained that it would: “first break down the allegedly infringed [computer] program into its constituent . . . parts”; then “sift out all non-protectable material, including ideas and expression that is necessarily incidental to those ideas”; and finally “compare[] the remaining creative expression with the allegedly infringing program.” App. 25 (internal quotation marks and citation omitted).

Using that framework, the court of appeals first held that the merger doctrine is inapplicable for two reasons: merger is “irrelevant” to copyrightability and Sun could have written the method headers in more than one way. App. 30–31. The court also rejected the district court’s reliance on the names-and-short-phrases doctrine. App. 33–35.

The Federal Circuit then held that Section 102(b)—which provides that “[i]n no case does copyright protection for an original work of authorship extend to any . . . system [or] method of operation,” 17 U.S.C. § 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.” App. 23 (internal quotation marks omitted; emphasis added). In the Federal

Circuit’s view, Section 102(b) serves only to codify the “idea/expression dichotomy”—the principle that “[c]opyright protection extends only to the expression of an idea—not to the underlying idea itself.” App. 18. Because “Google . . . could have designed its own . . . [application programming interface] packages if it wanted to do so,” and the method headers “could have been written and organized in any number of ways and still have achieved the same functions,” the court held that “Section 102(b) does not bar the packages from copyright protection.” App. 49. In the court of appeals’ view, “Section 102(a) and 102(b) are to be considered collectively so that certain expressions are subject to greater scrutiny.” App. 23.

The court of appeals also rejected the district court’s consideration of compatibility, calling it “[i]rrelevant to [c]opyrightability.” App. 50. According to the Federal Circuit, compatibility, and the fact that Java’s method headers “had become the effective industry standard,” are only factors to be balanced with others as part of a fair-use defense. App. 45–53, 57. The court remanded for a new trial on that defense. App. 53–62.¹

¹ The court of appeals addressed several other issues that are not relevant to the question presented in this petition. For example, the court affirmed the district court’s determination that Google copied “certain small snippets of code.” App. 102. By stipulation of the parties, the district court awarded no damages for that copying, which it characterized as “minor” and “innocuous.” App. 118, 120.

REASONS FOR GRANTING THE PETITION

The Federal Circuit’s decision warrants review for three reasons. *First*, it presents a longstanding, widely-recognized split in the courts of appeals. *Second*, the Federal Circuit’s holding is in conflict with decisions of this Court and contrary to the plain language of the Copyright Act. *Third*, whether copyright may be used to evade the limits on patent protection, in order to secure 95-year (or longer) monopolies, is an exceptionally important question. This Court already recognized the certworthiness of this question by granting review in *Lotus*. Since then, the circuit split has only deepened and the question has grown even more important as software has become ubiquitous in daily life.

I. The Courts Of Appeals Are In Disarray About The Application Of Section 102(b) To Software.

The Copyright Act provides that copyright protection subsists in “original works of authorship.” 17 U.S.C. § 102(a). But that protection does not extend to all elements of an original work. Section 102(b) specifies that “in 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 [original] work.” *Id.* § 102(b).

As the Federal Circuit and other courts of appeals have acknowledged, the circuits are deeply divided on how to construe Section 102(b). *See, e.g.*,

App. 23–24; *Lotus Dev. Corp. v. Borland Int’l, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995); *Computer Assocs. Int’l Inc. v. Altai, Inc.*, 982 F.2d 693, 705 (2d Cir. 1992); *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366 (10th Cir. 1997). Some courts follow the statute’s plain meaning, holding that Section 102(b) precludes copyright protection for all systems or methods of operation, including those in computer programs. *See, e.g., Lotus*, 49 F.3d at 815. Like the Federal Circuit, however, other courts have rejected the statutory text and held that Section 102(b) is merely a reminder of the dichotomy between ideas (which are not copyrightable) and expressions of ideas (which generally are). *See, e.g., Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1234 (3d Cir. 1986). In those courts’ view, a “method of operation” embodied in a computer program is copyrightable so long as its creator could have designed it in different ways. *See id.* at 1234 (internal quotation marks omitted).

1. *Lotus* exemplifies the plain meaning approach. That case concerned a spreadsheet program’s menu command hierarchy, which organized commands such as “print,” “copy,” and “quit” into more than 50 menus and submenus accessible by users. 49 F.3d at 809. The First Circuit held that the hierarchy was a “method[] of operation,” and was therefore excluded from copyright protection under Section 102(b)—regardless of whether the hierarchy (or the overall program) satisfied the originality requirement of Section 102(a) and regardless of whether there were other ways to write or structure the hierarchy. *Id.* at 815.

The First Circuit reasoned that a “method of operation’ . . . refers to the means by which a person *operates* something, whether it be a car, a food processor, or a computer.” *Id.* (emphasis added). Because the “menu command hierarchy provides the means by which users control and operate” the Lotus 1-2-3 program, the hierarchy was a method of operation excluded from copyright protection. *Id.* For that reason, it was “immaterial” that “Lotus developers could have designed the Lotus menu command hierarchy differently.” *Id.* at 816.

In determining whether an element of a computer program is a method of operation, the First Circuit also took into account compatibility (whether the element enables the program to interact with other software or hardware) and the lock-in effect (whether users have invested time and effort in learning how to use the method of operation). The First Circuit noted that the fact “[t]hat the Lotus menu command hierarchy is a ‘method of operation’ becomes clearer when one considers program compatibility.” *Id.* at 817. The court rejected as “absurd” Lotus’s theory that, “if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used.” *Id.* at 817–18.

The Sixth Circuit has similarly held that, “even if a work is in some sense ‘original’ under § 102(a), it still may not be copyrightable because [of] § 102(b),” which excludes original methods of operation from copyright protection. *Lexmark Int’l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 534 (6th Cir. 2004). That court explained that, although systems

and 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.” *ATC Distribution Grp., Inc. v. Whatever It Takes Transmissions & Parts, Inc.*, 402 F.3d 700, 707 (6th Cir. 2005) (internal quotation marks omitted).

The Sixth Circuit also held that the merger doctrine precludes copyright protection for elements of a computer program that are necessary for compatibility. *Lexmark*, 387 F.3d at 536. The court explained that, if there is only one practical way to express an idea, that expression is not entitled to copyright protection. *Id.* at 535. “Program 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.” *Id.* at 536 (internal quotation marks omitted).

2. Like the Federal Circuit in this case, other courts of appeals have disagreed with the First and Sixth Circuits in a number of respects. The Third Circuit, for example, insists that all elements of a computer program, including its structural elements, are copyrightable so long as the program could have been written differently and still served the same high-level purpose, such as “to aid in the business operations of a dental laboratory.” *Whelan*, 797 F.2d at 1238. In that court’s view, Section 102(b) “was not intended to enlarge or contract the scope of copyright protection,” only to reinforce the “somewhat metaphysical” dichotomy between idea and expression, with “idea” referring to a program’s general purpose. *Apple Computer, Inc. v. Franklin*

Computer Corp., 714 F.2d 1240, 1252, 1253 (3d Cir. 1983).

The Second Circuit has plowed a third path: the so-called “abstraction/filtration/comparison” test. Under that test, a court should first “dissect the allegedly copied program’s structure and isolate each level of abstraction contained within it.” *Altai*, 982 F.2d at 707. Then, the court should “filter[] . . . protectable expression from non-protectable material.” *Id.* 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 has distinguished its approach from the Third Circuit’s “inadequate . . . formulation that a program’s overall purpose equates with the program’s idea.” *Id.* at 705. The First Circuit, in turn, rejected the Second Circuit’s test, finding it “misleading” because “abstracting menu command hierarchies down to their individual word and menu levels and then filtering idea from expression at that stage . . . obscures the more fundamental question of whether a menu command hierarchy can be copyrighted at all.” *Lotus*, 49 F.3d at 815.

Like the Second Circuit, the Fifth and Tenth Circuits employ the abstraction/filtration/comparison test. See *Eng’g Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335 (5th Cir. 1994); *Eng’g Dynamics, Inc. v. Structural Software, Inc.*, 46 F.3d 408 (5th Cir. 1995) (supplemental opinion); *Gates Rubber Co. v. Bando Chem. Indus.*, 9 F.3d 823 (10th Cir. 1993). In adopting that test, the Tenth Circuit

expressly disagreed with *Lotus*, holding that “although an element of a work may be characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection.” *Mitel*, 124 F.3d at 1372. The court opined that Section 102(b), despite its plain text, does not withdraw copyright protection from methods of operation. Instead, “sections 102(a) & (b) interact to secure ideas for [the] public domain and to set apart an author’s particular expression for further scrutiny.” *Id.* That court thus “declin[ed] to adopt the *Lotus* court’s approach to section 102(b), and continue[d] to adhere to [its] abstraction-filtration-comparison approach.” *Id.*

3. In addition to disagreeing about whether to replace Section 102(b)’s plain language with one of the court-created standards discussed above, the courts of appeals have divided on related issues, including the relevance of compatibility to copyrightability. As noted above, the First and Sixth Circuits treat compatibility and lock-in as important if not dispositive considerations. The Second Circuit agrees with those circuits that “compatibility requirements of other programs with which a program is designed to operate” are relevant to copyrightability, as part of the “filtration” step of its abstraction/filtration/comparison test. *Altai*, 982 F.2d at 709–10. In contrast, the Third Circuit held that “compatibility with independently developed application programs . . . is a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged.” *Apple Computer*, 714 F.2d at 1253.

The courts of appeals are similarly divided on the merger doctrine. As noted above, the Sixth Circuit has split from other courts of appeals by holding that the merger doctrine precludes copyright protection for elements of a computer program necessary for interoperability. See *Lexmark*, 387 F.3d at 536. Other courts of appeals do not even agree that the merger doctrine limits copyrightability (in any way), holding that it is only an affirmative defense to infringement after copyrightability has been established—greatly diminishing its practical importance. See, e.g., *Kregos v. Associated Press*, 937 F.2d 700, 705 (2d Cir. 1991); see also pp. 28–29, *infra*.²

4. The decision below recognizes and deepens the circuit split. The Federal Circuit held that, under Ninth Circuit precedent: Section 102(b) does not exclude systems or methods of operation from copyright protection; a judicially-created abstraction/filtration/comparison test governs instead; “[i]nteroperability [a]rguments are [i]rrelevant to [c]opyrightability”; the merger doctrine does not restrict copyright protection for computer code necessary for interoperability so long as the original

² As the Federal Circuit recognized, the circuit courts’ disarray is so complete that they do not even agree on the correct standard of appellate review. App. 16 n.3. Compare *Matthew Bender & Co. v. West Publ’g Co.*, 158 F.3d 674, 681 (2d Cir. 1998) and *North Coast Indus. v. Jason Maxwell, Inc.*, 972 F.2d 1031, 1035 (9th Cir. 1992) (clear-error standard) with *Yankee Candle Co. v. Bridgewater Candle Co.*, 259 F.3d 25, 34 n.5 (1st Cir. 2001) and *Publications Int’l, Ltd. v. Meredith Corp.*, 88 F.3d 473, 478 (7th Cir. 1996) (*de novo* standard).

author could have written the code in more than one way; and merger plays no role in the copyrightability analysis in any event. *See* App. 23, 24, 50.

If the Federal Circuit’s view of Ninth Circuit precedent is correct, that circuit is in conflict with other circuits on all of those important points of law. If the Federal Circuit’s understanding of Ninth Circuit law is wrong, the Ninth Circuit is still in conflict with the courts on the other sides of the circuit splits. Either way, the longstanding division in lower court authority persists and requires this Court’s resolution.

II. The Federal Circuit’s Decision Runs Afoul Of The Statute, This Court’s Controlling Precedents, And The Distinction Between Patent And Copyright.

As explained above, the Federal Circuit held that the method headers are copyrightable even if they constitute, or embody, systems or methods of operation. App. 23; pp. 11–12, *supra*. That holding is wrong. It is contrary to the text of the Copyright Act, and it erases a fundamental boundary between patent and copyright law.

A. The statute codifies this Court’s exclusion of systems and methods of operation from copyright protection.

Under Section 102(a), an “original work of authorship” is generally copyrightable. Section 102(b) goes on to specify, however, that “in no case does copyright protection *for an original work of authorship* extend to any . . . system [or] 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) (emphasis added). There is nothing unclear or ambiguous about that provision. Though an original work of authorship is generally entitled to copyright protection, the protection afforded to that work does not extend to any systems or methods of operation included or embodied in the work. The statutory exclusion is explicit and absolute, governing “regardless of the form in which [a system or method of operation] is described, explained, illustrated, or embodied in such work.” *Id.*

The Federal Circuit opined, however, that “components of a program that can be characterized as a ‘method of operation’ may nevertheless be copyrightable.” App. 44. To reach that result, the court had to revise the statute, and it did: “Section 102(a) and 102(b) are to be considered collectively so that certain expressions are subject to greater scrutiny.” App. 23. The court did not explain whence this “greater scrutiny” test comes—it certainly does not come from the statutory text. The court did not explain what “greater scrutiny” means or how to apply it. Nor did the court even appear to apply greater scrutiny; it simply held that because Sun could have written the method headers in different ways, they were copyrightable. *See* App. 47.

The Federal Circuit’s error is especially stark because this Court determined more than twenty years ago that Section 102(b) “identifies specifically those elements of a work for which copyright is not available.” *Feist*, 499 U.S. at 356. The Court said nothing in *Feist* about replacing that specific,

statutory identification with a vague “greater scrutiny” test.

Ignoring this Court’s interpretation of Section 102(b), the court of appeals looked instead to the legislative history. App. 23. Legislative history can never displace clear statutory text. *See Milner v. Dep’t of Navy*, 131 S. Ct. 1259, 1267 (2011). And here, the legislative history specifically confirms that Section 102(b) means what it says: “processes or methods embodied in [a computer] program are not within the scope of the copyright law.” H.R. REP. NO. 94-1476, at 57 (1976).

The Federal Court pointed to a different passage in the legislative history that indicates, as this Court has explained, that Section 102(b) did not change preexisting law, “but merely clarified it.” *Feist*, 499 U.S. at 356; *see also* H.R. REP. NO. 94-1476, at 57; S. REP. NO. 94-473, at 54 (1975). That observation is fully consistent with the clear statutory text and the on-point legislative history quoted above. This Court had held, many decades before the 1976 Copyright Act, that systems and methods of operation (along with specific elements of expression that are “necessary incidents” to them) are not copyrightable. *Baker v. Selden*, 101 (11 Otto) U.S. 99, 103 (1880).

In *Baker*, Selden developed an accounting system and wrote a book explaining it. *Id.* at 100. He included in the book “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.” *Id.* Selden contended that “the ruled lines and headings, given to illustrate the

system, are a part of the book, and, as such, are secured by the copyright.” *Id.* at 101.

This Court rejected Selden’s argument; the forms were not copyrightable. The Court explained that “there is a clear distinction between the book, as such, and the art which it is intended to illustrate.” *Id.* at 102. “The copyright of a work,” in other words, “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.” *Id.* at 103 (emphasis added).

In light of that holding, the Federal Circuit’s decision runs headlong into not one, but two controlling decisions of this Court—*Feist* and *Baker*. The Federal Circuit attempted to distinguish *Baker* on the ground that it merely stands for a dichotomy between unprotectable ideas and protectable expression. App. 19. But nothing in *Baker* supports that interpretation. The case never even discusses that dichotomy. In any event, Section 102(b) codified *Baker* by unambiguously excluding systems and methods of operation from copyright protection, not by adopting a vague “greater scrutiny” test.

B. Systems and methods of operation are governed by patent, not copyright, law.

The Federal Circuit’s error is confirmed by the extent to which it would eliminate a fundamental distinction between patent and copyright law—and thus allow copyright to be used as an end-run around the limits on patent protection, including this Court’s recent decisions on patent-eligibility.

1. The *Baker* Court determined that the Patent Act, rather than the Copyright Act, governs the protectability of methods and systems. “The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself.” *Baker*, 101 U.S. at 105. “The object of the one is explanation; the object of the other is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters-patent.” *Id.*

Thus, under *Baker* and Section 102(b), copyright cannot be used to secure a monopoly on a system or method of operating something. “[T]he rules and methods of useful art have their final end in application and use; and this application and use are what the public derive from the publication of a book which teaches them.” *Id.* at 104. In the absence of a patent, “any person may practise and use the art itself.” *Id.*

For this reason as well, the Federal Circuit’s focus on whether there is more than one way to structure a system of method headers misses the point. There are, for example, many possible ways to design a keyboard, shorthand system, or accounting system. But under Section 102(b), *no* system or method of operation is protected by copyright.

2. Dismantling that boundary between patent and copyright protection would wreak havoc in the field of intellectual property by granting unwarranted, 95-year (or longer) monopolies on the basic building blocks of innovation. Unlike a claim to a copyright, “[t]he claim to an invention or discovery of an art or manufacture must be subjected to the

examination of the Patent Office before an exclusive right therein can be obtained; and it can only be secured by a patent from the government.” *Id.* at 102. The Patent Act imposes strict limits on patentability to ensure that a government-granted monopoly on *use* of an invention will serve its purpose of encouraging inventions and discoveries. *See, e.g.*, 35 U.S.C. §§ 101, 102, 103; *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 427 (2007).

Just last Term, this Court confirmed that, while some software-related patent claims may be eligible for patent protection under 35 U.S.C. § 101, many are not. *Alice Corp. Pty. Ltd. v. CLS Bank Int’l*, 134 S. Ct. 2347, 2358–59 (2014). Like Section 102(b) of the Copyright Act, Section 101 of the Patent Act protects future innovation by preventing anyone from “inhibit[ing] further discovery by improperly tying up the future use of the[] building blocks of human ingenuity.” *Id.* at 2354 (quoting *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1301 (2012)).

Extending copyright protection to methods and systems of operation would undermine the limits on patent protection. While the requirements for patentability are strict, Section 102(b) is the only requirement for copyrightability that does not present a very “low bar.” App. 17. Under Section 102(a), copyright protection is generally available for original works. The “originality requirement is not particularly stringent,” requiring “only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at

least some minimal degree of creativity.” *Feist*, 499 U.S. at 345, 358.

The threshold eligibility bar of Section 102(a) is so low as to be essentially non-existent for computer software, as confirmed by the Federal Circuit’s focus on whether Sun could have written the method headers in different ways. If one disregards the need to be compatible with other systems or programs, as the Federal Circuit did, there will nearly *always* be more than one way to write software code to accomplish a particular function (such as choosing the greater of two numbers), just as this sentence could have been written a dozen different ways without changing its import. Thus, virtually every element of every computer programming system or language would qualify for copyright protection under the court of appeals’ approach.

As *Baker* concluded, “[t]o give to the author of the [work] an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public.” 101 U.S. at 102. And a long-lasting fraud at that. Compared to the 20-year patent term, a copyright confers monopoly rights that can last for well over a century—for the remaining life of the author plus 70 years, for 95 years after first publication, or for 120 years after creation. 17 U.S.C. § 302. Permitting such an end-run around the carefully crafted limits on patent protection would stifle competition and innovation in the software industry—the very competition and innovation this Court has sought to protect by enforcing the

comparable limits on patentability. *See, e.g., Alice*, 134 S. Ct. at 2354.

That does not, of course, mean that all computer software is unprotected by copyright. There is no dispute, for example, that the implementing code that instructs a computer how to perform a method may be subject to copyright protection. *See* 17 U.S.C. § 101 (defining “computer program[s]” that may qualify as protectable works). But whether the method headers are entitled to protection is exclusively a question for patent law because the headers constitute, or embody, a system or method of operating the pre-written programs.

3. The Federal Circuit’s error is all the more glaring because it is essentially the same error for which this Court has repeatedly reversed the Federal Circuit in patent cases. The court of appeals criticized the district court for confusing “the threshold question of what is copyrightable—which presents a low bar—and the scope of conduct that constitutes infringing activity.” App. 17. It then transformed Section 102(b)’s limits on copyright eligibility into just one of several factors to be considered as part of a fair-use defense. *See* App. 50–56.

The Federal Circuit had similarly held that the limits on patent eligibility are minimal and that other requirements of the Patent Act do the real work in limiting monopoly protections. *See, e.g., Research Corp. Techs., Inc. v. Microsoft Corp.*, 627 F.3d 859, 869 (Fed. Cir. 2010) (referring to Section 101 of the Patent Act as a “coarse eligibility filter”). This Court has repeatedly corrected that

misperception in recent years, stressing the importance of enforcing Section 101's limits on patentable subject matter—including for software-related patents. *See, e.g., Alice*, 134 S. Ct. 2347; *Ass'n for Molecular Pathology v. Myriad Genetics, Inc.*, 133 S. Ct. 2107 (2013); *Mayo*, 132 S. Ct. 1289; *Bilski v. Kappos*, 130 S. Ct. 3218 (2010). But the Federal Circuit would now eviscerate the analogous limitation on copyright eligibility for some of the same types of works.

The Federal Circuit's error carries even more dire consequences in the copyright context than it did in the patent arena. There was at least a non-frivolous argument that the limits on patent eligibility were not exceptionally important because other limits on patentability could do some of the same work. *See, e.g., Mayo*, 132 S. Ct. at 1303–04 (rejecting the United States' argument to that effect). Here, such an argument would not even be colorable.

As discussed above, Section 102(b) places “any idea, procedure, process, system, method of operation, concept, principle, or discovery” in the public domain, as a matter of law, by excluding it from the scope of copyright protection. In contrast, the fair-use defense applies to materials that are within the scope of copyright protection, but blesses unauthorized uses that satisfy a multi-factor balancing test. *See* 17 U.S.C. § 107; App. 58–60. The Federal Circuit underscored the difference between the two by indicating that compatibility and lock-in are, in its view, not even the most important factors for a jury to consider as part of the fair-use inquiry. *See* App. 68.

In *Lotus*, the district court concluded, based on the facts of that case, that the defendant's use of the menu command hierarchy was not a fair use. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223, 240–45 (D. Mass. 1993), *rev'd*, 49 F.3d 807 (1st Cir. 1995). As the First Circuit recognized, however, Section 102(b)'s exclusion of the hierarchy from copyright protection controlled the outcome, making consideration of fair use unnecessary. 49 F.3d at 819.

C. The Java method headers are a system or method of operation.

This case illustrates the importance of applying Section 102(b) as written. The Java method headers, which enable programmers to use the familiar shorthand commands based on them, are certainly a system or method of operating the pre-written programs of the Java language and platform.

“All agree that Google was and remains free to use the Java language itself.” App. 108. That language is made up of “keywords and other symbols” as well as methods, “a set of pre-written programs to carry out various commands.” App. 106. As discussed above, programmers call the pre-written methods with shorthand commands that work only in software platforms that use the Java method headers. *See* pp. 6–8, *supra*.

The Second Circuit (including Learned Hand) long ago recognized that there is no “exclusive right to the use of a published system of shorthand.” *Brief English Sys., Inc. v. Owen*, 48 F.2d 555, 556 (2d Cir. 1931). Under *Baker*, a “system of condensing written

words into less than the number of letters usually used to spell them out” could be protected, if at all, only “by letters patent and not by copyright.” *Id.* (Under the Federal Circuit’s approach, presumably that case would have come out differently because there is more than one imaginable system of English shorthand; that absurd result illustrates how far from *Baker* the Federal Circuit strayed.)

As Oracle’s then-CEO Larry Ellison testified, moreover, “[t]he [Java] APIs are a command structure.” C.A. App. 20457. If Google had not replicated the method headers exactly, code that used the shorthand commands based on those headers would not have run on Android. *See pp. 9–10, supra.* Google took pains to replicate only the elements necessary to allow programmers to use the shorthand commands (*i.e.*, it copied only the method headers)—not the code that actually implements or performs the methods. App. 109. Computer programmers’ investment of time and resources in learning the shorthand commands confirms that the corresponding method headers, from which the shorthand commands are derived, constitute or embody the system or method of operating the pre-written programs of the underlying platform.

Compatibility and lock-in concerns confirm the applicability of both Section 102(b) and, in the alternative, the merger doctrine. If one must use specific computer code in order to operate computer programs such as the pre-written programs at issue here, that means, almost by definition, that the copied code is part of a system or method of operating the programs. *See, e.g., Lotus*, 49 F.3d at 817–18. As

discussed above, Google replicated the method headers so that computer programmers could operate the pre-written programs using the familiar shorthand commands derived from the headers. If Google changed the headers, the commands would not successfully operate the methods.

Copyright cannot lock up this system or method of operation any more than it could lock up the QWERTY keyboard. Pressing a key on a QWERTY keyboard sends a command that causes a computer to perform a specific function, such as drawing a “Q” on the screen. QWERTY is thus both a keyboard design and a command structure for causing computers of all kinds to produce letters and symbols—just as the method headers are the command structure for using the pre-written programs in the Java and Android platforms.

Oracle and the Federal Circuit have emphasized that, because Google replicated the method headers from only 37 of the Java packages, programs written in Java for the Java platform will not necessarily run as intended on the Android platform. App. 56–57. As the district court observed, however, “imperfect interoperability, and Oracle’s angst over it,” only prove the point by “illustrat[ing] the character of the command structure as a functional system or method of operation.” App. 160.

There is no dispute that Google replicated the method headers that were most important for mobile devices precisely because of the lock-in effect: like computer users who are familiar with the QWERTY keyboard layout, programmers were already accustomed to using the Java shorthand commands

based on the headers. App. 58. Google’s decision *not* to use more than it needed for a mobile-device platform certainly does not *expand* the scope of Oracle’s copyright protection, any more than a decision to omit the number keys on a keyboard would make a copyright claim for QWERTY more plausible.

Indeed, this case is a prime example of the importance of compatibility and lock-in. Programmers have invested significant time and effort in learning the Java language, including the shorthand commands. *See* p. 2, *supra*. But now, long after Sun lured computer programmers into the Java community and after any patent protection likely would have expired, Sun’s successor Oracle is attempting to build a wall around use of Java’s method headers. That would work precisely the “surprise and . . . fraud” on the public that *Baker* sought to prevent. *See* 101 U.S. at 102.

III. This Case Presents a Recurring Question of Exceptional Importance.

This case is an ideal vehicle for considering the question this Court tried to resolve in *Lotus*. As discussed above, this case alone is exceptionally important, as it involves both a ubiquitous interface (the method headers of the Java programming language) and a product relied on by many millions of people daily (the Android platform).

Moreover, the district court’s detailed factual findings and the Federal Circuit’s legal analysis cleanly present the question presented. Although the Federal Circuit remanded for a retrial on fair use,

the court of appeals definitively resolved the threshold legal question presented in this petition. There is no need to await a second trial on fair use before considering that question—especially considering the pressing need for this Court’s resolution.

The decision below is casting a pall over computer hardware and software development. *See, e.g.,* Van Lindberg, *The Copyrightability Of APIs In The Land Of OpenStack* (2014), available at <http://www.rackspace.com/blog/the-copyrightability-of-apis-in-the-land-of-openstack/>. As history has shown, the ability to build on existing interfaces in creating new products and services is a critical driver of innovation in the computer and software fields.

When IBM created the personal computer, for example, it developed an interface called the Basic Input/Output System. Competitors like Compaq and Phoenix re-implemented that system to create their own IBM-compatible computers, increasing the number of choices available to consumers. *See* Charles H. Ferguson & Charles R. Morris, *Computer Wars: The Post-IBM World* 53–55 (1994). Later, Apple used the pre-existing UNIX application programming interface in its computers’ operating system, allowing programmers familiar with UNIX to write software that could run on Apple’s innovative computers. *See* Joe Wilcox, *Will OS X’s Unix Roots Help Apple Grow?*, CNET.com (May 21, 2001). Oracle built upon the Linux operating system in much the same way. *See generally* Oracle Corp., *Frequently Asked Questions, Oracle Linux* (2014), available at <http://www.oracle.com/us/technologies/>

027617.pdf. And in order to compete in the word-processing field, Microsoft re-implemented WordPerfect's interface so that Microsoft Word, a competing product, could open documents created in WordPerfect. Br. of *Amici Curiae* Rackspace US, Inc. *et al.*, at 12–13, *Oracle Am., Inc. v. Google Inc.*, Dkt. No. 116, Nos. 13-1021, *et al.* (Fed. Cir. May 30, 2013). As these examples show, innovation depends on software developers' ability to achieve compatibility with, and build on, what has come before as they create new products and services.

The need to use existing interfaces without fear of copyright liability is even more essential in today's interconnected world. Cloud computing, for example, allows users to access virtual storage facilities and processing power from anywhere in the world via the Internet. *See* Amazon Web Services, *AWS Products & Solutions* (2014), available at <http://aws.amazon.com/>. Because the major cloud computing providers (Amazon, Eucalyptus, and CloudStack) use compatible interfaces, consumers are able to switch platforms and services seamlessly regardless of which browser or operating system they use. Steven J. Vaughan-Nichols, *OpenStack vs. CloudStack: The Beginning of the Open-Source Cloud Wars*, ZDNet (Apr. 12, 2012), available at <http://www.zdnet.com/blog/open-source/openstack-vs-cloudstack-the-beginning-of-the-open-source-cloud-wars/10763>. Those services compete with each other to provide the best implementations of the cloud-services interface; none should be entitled to an exclusive right to use the method of operation itself.

To take another example, millions of people use a computer program called Wine to make Microsoft Windows programs run on different operating systems. Wine works by re-implementing the Windows interface so that Windows programs will run on other operating systems. WineHQ, *About Wine*, available at <http://www.winehq.org/about/>. If Microsoft could threaten Wine with copyright liability, Wine could be shut down, depriving its customers of the ability to run Windows-based software on their computers.

Domestic and international laws also reflect the importance of protecting the public's right to use interfaces freely, without risking copyright liability. Congress has authorized "reverse engineering" for the "purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs." 17 U.S.C. § 1201(f). The European Union's Software Directive similarly provides a broad exception from liability for "black box reverse engineering." Council of Ministers Directive 91/250/EEC of 14 May 1991 on the Legal Protection of Computer Programs, Art. 5(3), 1991 O.J. (L 122).

Those laws make sense because, after identifying and analyzing the computer code that is necessary to achieve interoperability, developers are free to use it, as Google did here. Indeed, the European Union's highest court recently held that "neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit

certain of its functions constitute a form of expression of that program and, as such, are not protected by copyright.” *SAS Institute Inc. v. World Programming Ltd.* Case C-406/10 ¶ 71, 2012 E.C.L.I 259, [2012] 3 C.M.L.R. 4. A contrary conclusion would “amount to making it possible to monopolise ideas, to the detriment of technological progress and industrial development.” *Id.* ¶ 40.

As these real-world examples and laws reflect, the developer community has long understood that interfaces are free for everyone to use. That understanding has enabled all of the innovation described above, and much more. The Federal Circuit’s decision turns this understanding on its head, balkanizing computer languages and interfaces, requiring programmers to build from the ground up, precluding interoperability, and depriving consumers of the benefits of compatibility. At a bare minimum, that would make innovation much costlier and raise severe barriers to entry.

The decision below also inflicts particular and immediate hardship on smaller companies and start-ups—major sources of jobs and innovation. See Tim Kane, Ewing Marion Kauffman Foundation, *The Importance of Startups in Job Creation and Job Destruction* 3 (2010). These start-ups (the ranks of which Google, Sun, and Oracle once were members) are characterized by extraordinary creativity. They are innovating all the time, building on existing technology to bring products and services to market. To attract customers, these new market entrants must build on what has come before.

Consider how difficult it would have been for Tesla to build an electric car if the familiar arrangement and functions of a steering wheel, accelerator, and brake pedal were protected. The Java method headers and shorthand commands derived from them are to today's software programmers as those standard controls are to today's drivers—crucial methods for operating a complex system.

Delay in resolving this issue would magnify the harm caused by the decision below by impairing important innovation now in the fast-moving, high-technology sector. Just last Term, this Court granted review of an important copyright case even though there was no circuit split, and barely any percolation in the courts of appeals, because of the need for a timely ruling. *See Am. Broad. Cos. v. Aereo, Inc.*, 134 S. Ct. 2498 (2014). This case is no less important, as confirmed by the filing of eleven *amicus* briefs by dozens of *amici* (on both sides) in the court of appeals. Especially considering the clear and well-recognized circuit split on this issue, and the fact that this Court has already recognized the issue's certworthiness by granting review in *Lotus*, the Court should resolve this important and pressing issue now.

CONCLUSION

The petition for a writ of certiorari should be granted.

Respectfully submitted.

BRUCE W. BABER
KING & SPALDING LLP
1180 Peachtree Street, NE
Atlanta, GA 30309

ROBERT A. VAN NEST
STEVEN A. HIRSCH
CHRISTA M. ANDERSON
MICHAEL S. KWUN
DAN JACKSON
KEKER & VAN NEST LLP
633 Battery Street
San Francisco, CA 94111

DARYL L. JOSEFFER
Counsel of Record
ASHLEY C. PARRISH
ADAM M. CONRAD
ETHAN P. DAVIS
KING & SPALDING LLP
1700 Pennsylvania Ave., NW
Washington, DC 20006
(202) 737-0500
djoeffe@kslaw.com

RENNY HWANG
GOOGLE INC.
1600 Amphitheatre Parkway
MOUNTAIN VIEW, CA 94043

Counsel for Petitioner