No. 18-956

IN THE

Supreme Court of the United States

GOOGLE LLC,

Petitioner,

v.

ORACLE AMERICA, INC.,

Respondent.

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

BRIEF IN OPPOSITION

Dorian E. Daley
Deborah K. Miller
Matthew M. Sarboraria
ORACLE AMERICA, INC.
500 Oracle Parkway
Redwood Shores, CA
94065

Dale M. Cendali
KIRKLAND & ELLIS LLP
601 Lexington Avenue
New York, NY 10022

E. Joshua Rosenkranz
   Counsel of Record
Annette L. Hurst
ORRICK, HERRINGTON &
 SUTCLIFFE LLP
51 West 52nd Street
New York, NY 10019
(212) 506-5000
jrosenkranz@orrick.com

Counsel for Respondent

QUESTIONS PRESENTED

The Copyright Act protects "original works of authorship," including "literary works," 17 U.S.C. §102(a), defined as "works … expressed in words, numbers, or other verbal or numerical symbols or indicia," *id.* §101. "Computer program[s]," which the Act defines as "set[s] of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result," are literary works copyrightable under the Act. *Id.* Google copied 11,500 lines of Oracle's original, human-readable computer source code, as well as the intricate structure and organization of 37 large packages of computer programs, into a competing software platform. The questions presented are:

1. Does the Copyright Act protect Oracle's computer source code that Google concedes was original and creative and that Oracle could have written in any number of ways to perform the same function?

2. Was the Court of Appeals correct in holding that it is not fair use as a matter of law for Google to copy Oracle's code into a competing commercial platform for the purpose of appealing to Oracle's fanbase, where Google could have written its own software platform without copying, and Google's copying substantially harmed the actual and potential markets for Oracle's copyrighted works?

CORPORATE DISCLOSURE STATEMENT

Respondent in this Court, plaintiff-appellant below, is Oracle America, Inc. Oracle America, Inc. is not publicly traded. It is a subsidiary of Oracle Corporation (NYSE: ORCL), a publicly traded company. No other publicly held company owns 10% or more of its stock.

TABLE OF CONTENTS

Page

iv

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes & Rules

Other Authorities

INTRODUCTION[1]

Half of Google's petition has already been rejected, and the other half does not even purport to present a circuit conflict.

Google previously sought certiorari, in this very case, regarding whether Oracle's creative computer code is copyrightable. This Court denied review of that question in 2015 after inviting the views of the Solicitor General. The United States explained that Oracle's code is copyrightable and that Google's claim of a circuit conflict was meritless. Google now seeks review of that same question with the same arguments. But Google identifies nothing that has changed. The question has not recurred. Nor has software development suffered the devastating impact Google predicted; the industry is doing better than ever.

Google tacks onto its failed copyrightability arguments a request to review the Court of Appeals' fair use ruling. But Google does not allege a split; it just wants this Court to reassess how to apply the statutory four-factor test to this factual record.

The Court of Appeals' decisions correctly applied settled copyright doctrine to Google's egregious conduct. Oracle spent years and hundreds of millions of dollars writing a blockbuster work—a software platform. Google then refused Oracle's offer of a license

---

[1] "C.A." refers to the Court of Appeals Appendix. Amicus briefs are cited as "___ Br." The Government's previous invitation brief is cited as "U.S. Br." It is available here: <https://tinyurl.com/yyefhw7r>.

and copied the most recognizable portions of that work into a competing platform for the express purpose of capturing Oracle's fan base. Naturally, it inflicted incalculable market harm on Oracle. This is the epitome of copyright infringement, whether the work is a news report, a manual, or computer software.

Nevertheless, Google argues that wholesale copying must be allowed for the specific category of computer code it copied, which it describes as "lines of computer code that allow developers to operate prewritten libraries of code used to perform particular tasks." Pet. I. Joined by the usual amici who favor weaker protection for software, Google claims the Court of Appeals' decision imperils the future of "interoperable" software. But Google has conceded that it purposely made its platform *in*compatible with Oracle's. So this is no case to consider the copyright implications of interoperability. In any event, the Copyright Act's plain text protects code whether it is "used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. §101. Google cites not a single case—in any court—that has ever held that copying this volume of code (or this much structure and organization) into a competing work is fair.

It is therefore unsurprising that, in addition to the United States' support in this Court, numerous experts and stakeholders—including the former Register of Copyrights, the Business Software Alliance, other technology companies, and computer scientists—supported Oracle's position below. Microsoft, EMC Corporation, and NetApp warned that *Google's* position on copyrightability "would destabilize the

software industry." *Oracle I*, Microsoft Br. 9. These observers refute Google's doomsday predictions about the imminent demise of the software industry necessitating immediate review. Google's predictions also fail on their own terms because they are based on distortions of the Court of Appeals' holdings.

The petition should be denied.

STATEMENT OF THE CASE

Oracle Develops Computer Programs That Help Programmers Write Their Own Applications

Oracle created and continuously improved the Java 2 Standard Edition Platform ("Java SE" or "Java platform").[2] App. 4a-5a. The Java platform is one of the most popular and revolutionary works of computer software. App. 6a. It makes it easier to develop and run popular applications ("apps") written in the Java programming language. App. 5a-6a.

Two features contributed most to the Java platform's popularity. The first was that, unlike other platforms, it enabled app programmers to "write programs that 'run on different types of computer hardware without having to rewrite them.'" App. 4a. Hence, Java's credo: "write once, run anywhere." *Id.*

Second, the Java platform contains thousands of prewritten programs. Programmers can use those programs to build their own apps "rather than write their own code to perform [certain] functions from

---

[2] Oracle acquired Sun. We refer to them collectively as "Oracle."

scratch." *Id*. Each program contains two categories of code: "declaring code" and "implementing code." App. 4a-5a. The declaring code is like the chapter headings and topic sentences of an elaborate literary work: It introduces, names, and describes each prewritten program to help programmers learn and remember what those prewritten programs do. *Id*.

The URLConnection program, for example, has the following declaring code:

> *public URLConnection openConnection() throws java.io.IOException.*

A later app programmer who wanted to connect her application to BankofAmerica.com without writing her own code would call on Oracle's prewritten code by typing:

> *new URL('http://www.bankofamerica.com').open Connection().*

Then, when the program runs, the Java platform recognizes the code and invokes the corresponding "implementing code" to connect to www.BankofAmerica.com. App. 5a.

The prewritten programs are interconnected through an elaborate "structure, sequence, and organization." *Oracle II*, C.A. 50,959 (stipulation). At the highest level, related programs are organized into memorable groups called "packages." App. 4a. Oracle further organized each package into "a collection of classes and each class contains methods," which "perform[] a specific function," such as opening an internet connection. *Id*. These software packages are called

"APIs" ("application programming interfaces"). The Java platform contains over 30,000 methods, 3000 classes, and 166 packages. App. 5a.

Below is a map of the structure and organization of just the packages and classes (omitting the added layer of complexity of 30,000 methods). Each line represents the Java developers' conscious choice to create interrelationships among the packages and classes:
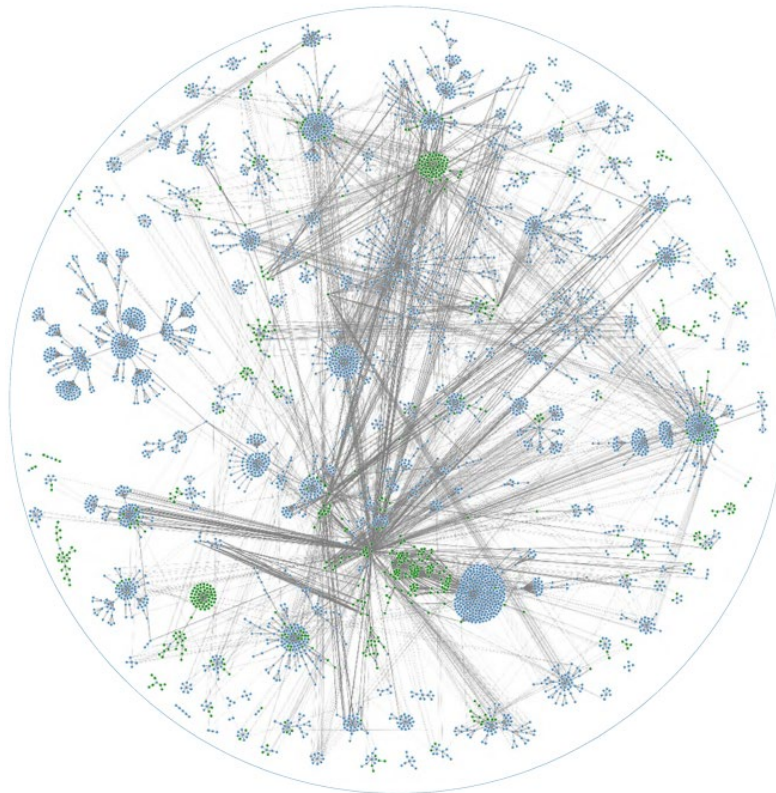
Oracle II, C.A. 1821. Given this vast complexity, Google acknowledged that "designing the Java API packages was a creative process." App. 140a.

All agree that Oracle had an infinite range of possible options to choose from in writing the code and organizing the programs in and among packages. App. 150a. It took Oracle's "most senior[,] experienced and talented" developers years to write some of the packages. *Oracle I*, C.A. 20,459, 20,791, 20,921.

Contrary to Google's keyboard analogy (at 5), the declaring code is far more expressive than the letters on a keyboard. First, each key on a keyboard is a single character, while declaring code can be "extremely long" and expressive, such as:

> *public abstract void verify (PublicKey*
> *key, String sigProvider)*
> > *throws CertificateException,No-SuchAlgo-*
> > *rithmException, InvalidKeyException,*
> *NoSuchProviderException, SignatureException*

Oracle II, C.A. 51,452-63. Second, while "ASDFG" expresses nothing to anyone, Oracle's declaring code communicates to programmers what each program does, how it relates to the other programs, and what you need to do to make it work. Third, in terms of complexity, there is no comparison between the relationship among 26 keys and the Java platform's intricate organization (depicted only partially above) among 30,000 programs.

Oracle Develops A Licensing Regime To Ensure Compatibility

Implicit in the foregoing explanation is a crucial distinction that Google elides—between *app* programmers who use Oracle's Java platform to create apps

and *platform* developers who copy it to commercialize their own competing platform.

App programmers write the sorts of fun, useful, and quirky apps that run on your tablet or smartphone. Oracle offers app programmers a license to use the platform for free. App. 5a.

Oracle recoups its investment in the Java platform mainly by licensing it to (1) hardware manufacturers who copy the platform onto their devices (e.g., PCs, phones, or tablets) to run the wide variety of apps from app programmers and (2) competing platform developers who want to use Oracle's programs to commercialize their own platforms. Any platform developer that does not want to take a license is free to develop its own platform with identical functions without copying the Java platform. Apple and Microsoft did it. App. 149a & n.5. But if a commercial platform developer wants to copy or adapt the Java platform, Oracle requires it to take a license.

One option is an "open source" license called OpenJDK that allows a developer to use the software packages for free so long as it agrees to make its revisions to the platform public and "give away those changes for free" to their competitors. App. 5a-6a. Commercial enterprises disfavor this option because they do not want to invest in making improvements only to give them away to their competitors. If the platform developer wants to keep its code proprietary, then it takes a commercial license and pays royalties to Oracle. *Id.* "To preserve the 'write once, run anywhere' philosophy, Oracle imposes strict compatibility requirements on [its commercial] licensees." App. 5a.

Under this licensing scheme, Java became "the leading platform for developing and running apps on mobile phones." App. 6a. Before Android, every company that wanted to use the Java platform took a commercial license, *see, e.g.*, *Oracle II*, C.A. 51,395-96, 51,411-14, 51,428-29, including smartphone manufacturers Blackberry, Nokia, and Danger, App. 35a, 50a. SavaJe—a company dedicated to building a specialized mobile phone platform—took a license. *Id*. Other companies, like IBM and Oracle (before acquiring Sun), who wanted to copy only the declaring code and structure and organization, also took a license. *See Oracle I*, C.A. 20,466-67.

Google Copies Thousands Of Lines Of Code And The Structure And Organization Of 37 Software Packages

In 2005, Google faced an existential threat. People with mobile devices were not using Google's search engine, causing Google to lose significant advertising revenue. It needed to quickly develop a platform tailored to mobile devices that would promote Google search. *Oracle II*, C.A. 54,336-38. Google believed success turned on attracting Java programmers to build apps for it. App. 6a. Google thought the alternatives "all suck[ed]" and knew it "[m]ust take [a] license" if it wanted to copy the Java APIs. *Oracle II*, C.A. 54,008, 54,012. But in license negotiations with Oracle, Google rejected the condition Oracle demanded of all commercial licensees: make Android "compatible with the Java" platform and "interoperable with other Java programs." App. 128a.

Instead, without any license, Google copied thousands of lines of Oracle's declaring code and the structure and organization of the 37 API packages it considered "key" to attracting Java mobile-app developers. App. 7a, 129a. Google also undermined "write once, run anywhere" by deliberately making Android *in*compatible with the Java platform, meaning Android apps run *only* on Android devices and Java apps do not run on Android devices. App. 130a.

Android's founder testified that Android became an overnight "competitor" to Oracle "targeting the same industry with similar products." *Oracle II*, C.A. 50,844; *see* App. 50a-53a. Android generated over $42 billion for Google from 2007 to 2016. App. 7a. "[M]any [Oracle] customers switched to Android," while those who stayed with Oracle "leverage[d]" Android for "steep discount[s]." *Id.* In short, Android "replaced" Java and "prevented" Oracle from licensing and competing in the developing smartphone market. App. 53a.

The Court Of Appeals Finds Oracle's Software Packages Copyright Protected, And This Court Denies Certiorari

Oracle sued Google for copyright infringement. App. 1a. The jury found Google infringed Oracle's copyright but hung on fair use. App. 130a-131a. After trial, the district court acknowledged that the declaring code and the structure and organization were "creative" and "original" but nevertheless held the declaring code was not copyrightable under the merger doctrine and the structure and organization was an uncopyrightable method of operation. App. 141a, 165a-166a.

The Court of Appeals reversed, holding both Oracle's declaring code and the structure and organization copyright-protected. App. 123a. The court observed that it is "well established that copyright protection [for computer programs] can extend to" both their "source code" and their "structure[] and organization." App. 139a. The court found it "undisputed that Google copied [thousands of] lines of declaring code and generally replicated the overall structure, sequence, and organization of Oracle's 37 … packages," App. 134a, and "undisputed that the declaring code and the structure and organization of the … packages are original," App. 140a.

The court first rejected Google's argument that the original expression of the lines of code "merge[d]" with unprotectable ideas. App. 150a-152a. There could be "no merger" of the code's expression with its ideas because Oracle's developers had "unlimited options" in writing and organizing the declaring code. App. 150a-151a. Next, the court rejected Google's argument that the structure and organization is an unprotectable "method of operation" under §102(b), finding it contrary to the statutory text and this Court's precedent. App. 158a-166a.

Google petitioned for certiorari. Upon invitation from this Court, the United States opposed Google's petition. It explained that Google's §102(b) argument "lack[ed] merit" and that no court has adopted it. U.S. Br. 10. It also demonstrated that there was no circuit split on copyrightability. *Id.* 19-22. This Court denied the petition.

After A Retrial, The Court Of Appeals Finds Google's Copying Was Not Fair Use As A Matter Of Law

On retrial, the jury concluded that Google's use was a fair use. *See* 17 U.S.C. §107. Oracle appealed, and the Court of Appeals reversed. App. 35a n.7. It "assume[d] that the jury resolved all factual issues relating to the historical facts in favor of the verdict" and carefully analyzed each of the four fair-use factors in light of those historical facts. App. 23a.

In a record-intensive analysis, the court concluded that factor 1 (purpose and character of the use) weighed against fair use: Google's use was "overwhelmingly commercial," App. 25a-28a, and it was not "transformative" both because (1) Google "made no alteration to the expressive content or message of the copyrighted material," and (2) Google used it for the "same … purpose" in the same "smartphone[] … context" that the Java platform was already in, App. 28a-37a.

The court found the second factor (nature of the copyrighted work) weighed in Google's favor. App. 40a-43a. The third (the amount taken) was "at best" neutral or favored Oracle, given that the code was a highly valuable part of the Java platform. App. 43a-47a.

The fourth factor (effect on existing and potential markets), which Google argued was the most important factor, "weigh[ed] heavily in favor of Oracle" in light of the unrefuted evidence that Android caused Oracle to lose customers and impaired Oracle's ability to "license its work for mobile devices." App. 47a-53a.

The Court found undisputed evidence that "Android … had a direct market impact" on Java SE in the markets for mobile phones and tablets, including smartphones. App. 51a. The court also found that, to the extent smartphones might be considered a potential market, it was a market that "undisputed evidenced showed … Oracle intended to license Java" in and Android "harm[ed]." App. 52a.

In sum, the factor Google identified as most important (the fourth) and the first "weigh heavily" in favor of Oracle, the third was neutral or favored Oracle, and the second favored Google. App. 53a. "Weighing these factors together," the panel "conclude[d] that Google's use of the declaring code and [structure and organization] of the 37 API packages was not fair as a matter of law." App. 53a-54a. Given Oracle's overwhelming showing on the first and fourth factors, any other outcome would have impermissibly created a *sui generis* category of fair use for software, effectively depriving it of copyright protection contrary to congressional mandate.

REASONS TO DENY CERTIORARI

I.   The Court Of Appeals' Copyrightability Decision Is Still Not Cert-Worthy.

Google's first question presented is "[w]hether copyright protection extends to a software interface." "Software interface" is a term Google invented for its petition; it is not defined in the Copyright Act, the record, the Court of Appeals' opinions, or any of the cases Google cites. It is as meaningless as asking whether protection extends to a "verbal interface." To know

whether copyright protection extends to a particular work, one needs to be precise about what that work is.

The work here is a set of 37 intricate packages of thousands of computer programs comprising one of the most creative and popular computer platforms ever written. *Supra* 3-6. Google copied 11,500 lines of that code, each consisting of words and symbols that express and communicate certain information to the reader, as well as Oracle's intricate organization of the programs. The question is whether *that* work is totally devoid of copyright protection.

The Copyright Act protects "original" works of authorship. 17 U.S.C. §102(a). A work is "original" if "it possesses at least some *minimal degree* of creativity." *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991) (emphasis added). As Google has conceded, App. 141a, the highly creative code and intricate organization that Google copied here exceed *Feist*'s low bar. This Court should deny review on Question 1, as it did before, because (A) there is no circuit split; (B) the Court of Appeals was right; and (C) there is a fatal vehicle defect.

A. This case presents no circuit split.

We begin with first principles: Google "nowhere disputes" that it is "well established" that original computer code and a program's structure and organization warrant copyright protection. App. 139a (discussing cases); *see* Pet. 3. That is because the Copyright Act covers "literary works," 17 U.S.C. §102(a), defined as works "expressed in words, numbers, or other verbal or numerical symbols," *id.* §101;

see U.S. Br. 4. Moreover, because "copyright protection extends beyond a literary work's strictly textual form to its non-literal components" (such as a detailed plot), the structure and organization of a computer program is also copyrightable. *Comput. Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 701-02 (2d Cir. 1992). Thus, when the Act declares that "Copyright protection subsists … in original works," including "literary works," it covers computer programs as long as they are original.

Despite this settled law, Google argues that courts are "deeply divided" about the copyrightability of the specific sorts of computer programs it copied here. Pet. 14. But Google cites no case holding thousands of lines of concededly original, creative computer code devoid of copyright protection. Nor does it cite any case that has so much as suggested that the structure and organization of a computer program with tens of thousands of interrelated components lacks copyright protection. Google seeks exceptional treatment for the subset of code at issue here, which it describes as: "lines of computer code that allow developers to operate prewritten libraries of code used to perform particular tasks." Pet. I. But Google ignores the Act's definition of "[c]omputer program[s]" as "set[s] of statements or instructions to be used directly *or indirectly* in a computer to bring about a certain result." 17 U.S.C. §101 (emphasis added); App. 136a-137a.

Google offers nothing more than the same purported splits based on the same increasingly stale cases it presented unsuccessfully four years ago. And nothing has changed since 2015. As the Government explained then in recommending denial: "Nothing

about the declaring code at issue here materially distinguishes it from other computer code," and Google has "identified no genuine conflict of authority" on the copyrightability of computer programs, their source code, or their structure and organization. U.S. Br. 10.

1. As before, Google's main purported split is about how §102(b) applies to computer code—specifically, about the relationship between §102(a) and 102(b). Section 102(a) provides that "[c]opyright protection subsists … in original works of authorship," including computer programs. Section 102(b), on which Google fixates, provides: "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."

There is no mystery—or split—about what §102 achieves. For well over a century, this Court has adhered to the fundamental principle, first articulated in *Baker v. Selden*, 101 U.S. 99 (1879), that copyright "protection is given only to the expression of the idea—not the idea itself." *Mazer v. Stein*, 347 U.S. 201, 217 (1954). This Court has repeatedly explained that §102 "codifie[s]" and "restate[s]" the idea/expression distinction. *Golan v. Holder*, 565 U.S. 302, 328-29 (2012); *Feist*, 499 U.S. at 356. Indeed, the drafters of §102(b) drew key language directly from *Baker*, most notably the phrase Google features so prominently: "method of operation."

Viewed this way, §102(a) and 102(b) are complementary. Section 102(a) declares that if a "work" is "original," it receives copyright protection. And

§102(b) explains that just because a work receives copyright protection does not mean that the protection "extend[s]" beyond the author's expression to monopolize any "idea" (or "process," "method of operation," or other synonym in §102(b)) described or embodied in the work. App. 137a-138a. For example, "a book on how to build a bicycle may be eligible for copyright protection" that covers only the particular way in which the book is written; "that copyright does not include any exclusive right to practice the bicycle-building method that the book explains." U.S. Br. 12.

Google does not deny how the idea/expression dichotomy applies here. As the Court of Appeals explained, it means that Oracle cannot seek protection for the "idea of organizing functions of a computer program" or "the 'package-class-method' organizational structure in the abstract." App. 164a. But Oracle may, as it does here, "claim[] copyright protection only in its *particular way* of naming and organizing each of the 37 Java API packages." *Id.* (emphasis added).

Despite this well-settled understanding that §102 codifies the idea/expression dichotomy, Google offers a "diametrically opposite" reading of §102(b). Pet. 13. It argues that §102(b) *subtracts* from §102(a)—that a "work" that is protected under (a) loses protection under (b) just because it can *also* be described as a "method of operation" (or a "system," "concept," or "principle"). Pet. 12. Google asserts that "the courts of appeals are deeply divided" on that question. Pet. 14.

No circuit has ever adopted a reading that is inconsistent with—much less "diametrically opposite" to—this Court's conclusion that "Section 102(b) in no

way enlarges or contracts the scope of copyright protection." *Feist*, 499 U.S. at 356 (quoting H.R. Rep. No. 94-1476 at 57 (1976)).[3] The Government documented, and underscored, the consensus view that §102(b) merely "codifies the idea/expression dichotomy" and that Google "therefore is incorrect in suggesting … that a work could be both an original work … protectable under Section 102(a) and a 'method of operation' … under Section 102(b)." U.S. Br. 13 (quotation marks and alterations omitted).

In support of its purported split, Google cites two cases. It leads with *Lotus Development Corp. v. Borland International, Inc.,* 49 F.3d 807 (1st Cir. 1995), emphasizing that the case divided this Court 4-4, *see* 516 U.S. 233 (1996). But *Lotus* does not advance Google's position for three reasons. First, it did not involve the copying of computer code. The First Circuit repeatedly emphasized that the accused infringer "did not copy any of Lotus's underlying computer code." 49 F.3d at 810; *accord id.* at 809-10, 812. "The First Circuit took pains to distinguish" code from the work at issue in *Lotus.* U.S. Br. 20. Google never explains how a case that did not involve copying code could create a "circuit conflict over the copyrightability of software interfaces," which Google defines to include only "lines of computer code." Pet. I, 11. Second, *Lotus* was about a "menu … hierarchy" of simple one-

---

[3] *E.g.*, *Toro Co. v. R & R Prods. Co.*, 787 F.2d 1208, 1211 (8th Cir. 1986) (§102(b) "is nothing more than a codification of the idea/expression dichotomy"); *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 836 (10th Cir. 1993) (same); *Miller v. Universal City Studios, Inc.*, 650 F.2d 1365, 1368 n.1 (5th Cir. 1981) (same); *Warner Bros. Inc. v. Am. Broad. Cos.,* 720 F.2d 231, 240 (2d Cir. 1983) (same).

word commands, such as "print" and "copy." 49 F.3d. at 809. Oracle's code is far more intricate and creative. *See* App. 159a-160a. Third, there is no way to read *Lotus* as adopting any view (much less a contrary view) of how §102(b) treats computer code as a "method[] of operation." It said in no uncertain terms that the "Lotus 1-2-3 code is *not* a[n] uncopyrightable 'method of operation." 49 F.3d at 816 (emphasis added).

Google's argument depends on a single opaque sentence in *Lotus*: "We think that 'method of operation,' as that term is used in §102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer." *Id.* at 815. That line could not possibly have meant that anything that "operates something" loses copyright protection. That would eliminate protection for all works that serve a practical function, rendering all computer programs uncopyrightable "methods of operation." As the Government explained, it would be "[]natural to describe [any] computer code as a 'method of operation,'" but "the Copyright Act makes clear that [all] such code can be copyrightable" so long as it is original. U.S. Br. 10, 13-15; *see* App. 162a-163a; 17 U.S.C. §101 (defining "computer program" as "instructions" that cause a computer "to bring about a certain result"); *Oracle I*, Former Copyright Register Oman Br. 13 (Google's position "threatens to do violence to the very concept of copyright protection of software.").

In the 25 years since *Lotus*, no appellate court has applied *Lotus*'s statement about "methods of operation" the way Google proposes. U.S. Br. 20-21. And the First Circuit has since clarified that it agrees that

§102(b) "is a codification of the most fundamental axiom of copyright law that no author may copyright his ideas." *Situation Mgmt. Sys., Inc. v. ASP. Consulting LLC*, 560 F.3d 53, 55 (1st Cir. 2009) (quotation marks, alterations omitted); *accord Greene v. Ablon*, 794 F.3d 133, 156 (1st Cir. 2015).

The second opinion Google invokes, *Lexmark International, Inc. v. Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004), "provides even less support for" Google because it explicitly negates any circuit conflict, U.S. Br. 21. The Sixth Circuit agreed that §102(b) "embodies the common-law idea-expression dichotomy." 387 F.3d at 534. Google suggests that the court contradicted that explicit holding, in dicta, observing that "even if a work is in some sense 'original' under §102(a), it still may not be copyrightable because [of] §102(b)." *Id.* But in context, that simply means that, after assessing originality, courts must analyze whether the author is impermissibly attempting to extend its copyright to an entire idea, rather than limiting it to the author's particular expression of an idea. *Lexmark* nowhere embraced Google's view that original computer programs become unprotectable because they "operate[]" something. U.S. Br. 21.

Since *Lexmark*, the Sixth Circuit, too, has repeated that §102(b) codifies the idea/expression dichotomy. *See, e.g.*, *Stromback v. New Line Cinema*, 384 F.3d 283, 296 (6th Cir. 2004); *accord ATC Distrib. Grp., Inc. v. Whatever It Takes Transmissions & Parts, Inc.*, 402 F.3d 700, 707 (6th Cir. 2005) (quoted at Pet. 13).

Google asserts that three circuits have "adopted still another approach," which is less "categorical." Pet. 13. In fact, they all read §102 as a codification of the idea/expression dichotomy. *Supra* 17 n. 3 (collecting cases). What Google points to in these cases are discussions of a different question: *How* to distinguish idea from expression? Under the "'abstraction/filtration/comparison' test," courts "filter[]" unprotected ideas from protected expression before "compar[ing]" that expression to the allegedly infringing work. *See Altai*, 982 F.2d at 706-07. As Google notes, that is what the Court of Appeals did here. Pet. 14.[4]

2. Google briefly mentions the merger doctrine as the source of two further splits. Pet. 14-16. Merger is related to the idea/expression dichotomy. It establishes: "When the idea and its expression are … inseparable," because there is only one way to express that idea, "copying the expression will not be barred, since protecting the expression in such circumstances would confer a monopoly of the 'idea.'" *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1168 (9th Cir. 1977). Of the two circuit conflicts Google asserts, one is not a split, and neither is presented in this case. *See* U.S. Br. 22.

Google manufactures the first out of another stray line of dicta in *Lexmark*. *Lexmark* addressed so-called

---

[4] Google also cites *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1522 (9th Cir. 1992) and *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596, 603 (9th Cir. 2000), in support of its copyrightability argument. *See* Pet. 17. But "[b]oth *Sega* and *Sony* are fair use cases," not copyrightability cases, App. 166a, and are therefore addressed in that section (at 27-28).

"'lock-out' codes," tiny snippets of computer code that prevent certain components from working with other devices—there, "printers and toner cartridges." 387 F.3d at 536. *Lexmark* found the plaintiff was unlikely to prevail in showing the snippet—occupying less memory than "the phrase 'Lexmark International, Inc. vs. Static Control Components, Inc.'"—was sufficiently original to be copyrightable. *Id.* at 529-30, 539-41. By contrast, Oracle seeks to protect not a single "individual line[] of code" but thousands of lines of code and an intricate organization. App. 154a. Moreover, Oracle had "unlimited options" available in drafting, App. 150a, whereas the lock-out code in *Lexmark* was "dictate[d]" by "external constraints," leaving that programmer without "much choice" in drafting the code. 387 F.3d at 539-40 (quotation marks omitted).

Without mentioning that holding, Google fixates (at 15) on *Lexmark*'s passing suggestion that "[p]rogram code that is strictly necessary to achieve current compatibility presents a merger problem." 387 F.3d at 536 (citation omitted). It is unclear what the Sixth Circuit meant by this, but whatever it meant about the relevance of compatibility could not conflict with the decision here. The Court of Appeals found that Android is emphatically *not* compatible with Java. App. 172a-173a. Google intentionally designed Android *not* to be "interoperable with other Java programs." App. 128a-129a.

Google's other purported circuit split is over whether merger bears on copyrightability or instead is a defense to infringement. Pet. 15-16. But the Court of Appeals held that question does not matter here. For reasons discussed immediately below, it held that

"merger does not apply on [this] record," "[r]egardless of when the analysis occurs." App. 148a.

B. The Court of Appeals correctly found Oracle's software protected.

The Government's brief exhaustively explains why the Court of Appeals' copyrightability analysis is correct. U.S. Br. 10-19. Google has not overcome that analysis. Apart from pressing a reading of §102(b) that this Court has rejected, *supra* 15-17, Google's main merits argument is that the Court of Appeals misapplied the merger doctrine, which as explained above, holds that when there is only one way to express an idea, it is permissible to copy that expression.

Here, Google admitted and the district court found that Oracle could have written its code any number of ways. App. 165a-166a. As Oracle's code was not "preordained" and Oracle had "*unlimited* options as to the selection and arrangement of" the code and structure that Google copied, the Court of Appeals held merger did not apply. App. 150a (emphasis added); *see* App. 164a.

Google does not dispute that *Oracle* had infinite options when it wrote the platform. It argues the proper focus should be the "expressive choices available to *Google* when it" copied Oracle's code and "created Android." Pet. 20. Even here, Google concedes that it could have written a competing platform completely differently—as Apple and Microsoft did. *Supra* 7. Its contention is much narrower: that "using the Java API declarations was the only way to allow independent developers to rely on their preexisting

knowledge" of the Java APIs to "creat[e] new programs." Pet. 19.

As both the Court of Appeals and the Government have correctly recognized, Google's approach is doubly flawed. First, copyrightability depends on the choices that were available *to the original author* when it created its work. Copyright "subsists from its creation" and "endures" for a set term. 17 U.S.C. §302. It "protect[s] all works of authorship from the moment of their fixation in any tangible medium of expression." App. 151a (quoting CONTU Report at 21); *accord* U.S. Br. 18 n.2; *Oracle I*, Former Copyright Register Oman Br. 20. That was when Oracle wrote it.

Second, a work does not lose copyright protection just because it becomes so wildly popular that others see a huge benefit in copying it. By Google's logic, an author writing a novel about child wizards could copy from *Harry Potter* to "allow" readers "to rely on their preexisting knowledge of" JK Rowling's famous characters, fictional locations, and unique spells. The Java code and structure were not "the only and essential means of accomplishing [the] given task," Pet. 18, unless that task was to impermissibly hijack the expressive value of what Oracle wrote.

C. The petition has a fatal vehicle defect.

The question Google presents would have no effect on the outcome. As the Court of Appeals explained, "Oracle claims copyright protection with respect to both: (1) … declaring source code; and (2) … the structure, sequence, and organization of each of the 37 Java API packages." App. 139a-140a. Having separately analyzed each, App. 146a-157a, 158a-

166a, the Court of Appeals found *both* the lines of declaring code *and* the structure and organization copyright-protected and infringed, App. 123a, 136a.

Google's copyrightability question focuses exclusively on the lines of declaring code. *See, e.g.*, Pet. 12, 16, 19. It asks about the copyrightability of so-called "software interfaces," Pet. I, a term Google makes up and defines to include *only* "lines of computer code." *Id.* By its express terms, this question does not apply to the second, independent copyrightability holding supporting the judgment.

In fact, Google's petition never even mentions the copyrightability of structure and organization. Google's sole reference to structure and organization is the assertion that the declaring code "embodie[s]" the API packages' structure and organization. Pet. 8. But Google does not dispute that the structure and organization is entitled to copyright protection independent of the declaring code, App. 139a-140a, 158a-166a, or that Google would be equally liable if it had not copied a single line of code but had only duplicated the structure and organization of the Java API packages. Because this Court's review would not be outcome determinative, it should deny review.

II. Google's Request For Error Correction On Fair Use Does Not Warrant Review.

Google nowhere contends there is a circuit conflict on fair use. As with copyrightability, it cites not one case where a court has found it fair to copy this much code into a competing commercial product. Nor does it cite a single opinion assessing fair use where an infringer copied into its product "lines of computer code

that allow developers to operate prewritten libraries of code used to perform particular tasks." Pet. I. Instead, Google argues only that the "Federal Circuit misapplied the precedents of this Court and others," Pet. 21, and "improperly revisit[ed] … the jury's implicit factual determinations," Pet. 28. That is a naked plea for fact-bound error correction, which is no basis for this Court's review. Sup. Ct. R. 10.

Google agrees the Court of Appeals correctly stated the basics of fair use law. Pet. 22. Fair use is "a limited exception to the copyright holder's exclusive rights … 'for purposes such as criticism, comment, news reporting, teaching …, scholarship, or research.'" App. 13a (quoting 17 U.S.C. §107). Congress directed courts to engage in a "case-by-case determination" of fair use, *Harper & Row Publ'rs, Inc. v. Nation Enters.*, 471 U.S. 539, 549 (1985), guided by four factors: (1) the purpose and character of the use; (2) the nature of the copyrighted work; (3) the substantiality of copied material; and (4) the effect on the potential market for or value of the copyrighted work, 17 U.S.C. §107.

Google's petition skirts this inquiry. Though it purports to attack the manner in which the Court of Appeals applied the four factors, it does not go through the factors nor balance the results. It ignores half of factor 1 (that its use was overwhelmingly commercial); it skips factor 3; on factor 4, it ignores uncontradicted evidence of direct harm to Java from its copying; and it says not a word about the Court of Appeals' balancing of the factors.

Instead, Google claims the Court of Appeals made three discrete errors in applying these factors to the

particular facts of this case. Google's purported errors are meritless.

A. Google leads with an extension of its claim of software exceptionalism, the ever-so-tentative argument that the court adopted the wrong "overall approach" by "seemingly" "fail[ing] to adapt the fair-use doctrine to the functional nature of software interfaces." Pet. 23. But there is no dispute that Congress directed the same fair-use analysis for all works, from the most functional manual to the most creative novel. *See* 17 U.S.C. §107. Google cites no case holding that the traditional four-factor fair-use analysis should be ignored when it comes to software—or to the amorphous category Google calls "software interfaces."

That is not to say that each fair use *factor* dictates the same result for code as for a novel. As Google eventually acknowledges, there is a place in the traditional fair use analysis that considers the functional nature of the work: "the second fair-use factor, the nature of the copyrighted work." Pet. 23. Like every court that has addressed computer code, that is where the Court of Appeals addressed it. And what it said there explains why Google had to hedge: The court acknowledged that "*functional* considerations were both substantial and *important*" and, for that reason, found that factor 2 "favors a finding of fair use." App. 42a (emphasis added). If this case involved a novel, the court would have weighed factor 2 heavily against fair use. Treating Oracle's work differently from a novel was all about "adapt[ing] … to the functional nature of software."

The only way the court could have given more weight to the software's functional nature would have been to hold that factor 2 overrides all the others and *requires* a finding of fair use for computer code. App. 43a. Google waived any such argument by repeatedly arguing that factor 4 ("market harm") "is the most important factor" in this case. *Oracle II*, Google Br. 58, 68.

Regardless, Google is wrong in this repurposed effort to deprive software of copyright protection. It cites no case that has ever held that factor 2 is dispositive for software. Courts weigh factor 2 for software the way they do for any other work—on the facts of each case. *See Wall Data Inc. v. L.A. Cty. Sherriff's Dep't*, 447 F.3d 769, 780 (9th Cir. 2006) (finding factor 2 weighed against fair use in software case); *see also Harper & Row*, 471 U.S. at 569 (finding no fair use where factual work was copied); *Oracle II*, IP Professors Br. 27-29. The per se rule Google seeks would vitiate the statute's four-factor analysis; override the "case-by-case nature" of the fair use analysis, *Harper & Row*, 471 U.S at 549; and "effectively negate Congress's express declaration—continuing unchanged for some forty years—that software is copyrightable." App. 43.

The two Ninth Circuit software cases Google cites do not support its argument. Pet. 23-24. The Federal Circuit, which decided this case under Ninth Circuit law, recognized that these precedents would be binding if applicable. But it found them "materially" distinguishable because the balance of factors—most notably factor 1 (purpose of the use)—was different. App. 54a. In both, the accused infringer copied the

code only to *research* how it worked, with a view toward developing a non-infringing compatible product made with "entirely new … code." *Sony*, 203 F.3d at 606-07; *Sega*, 977 F.2d at 1525-26. Both cases emphasized the centrality of the fact that the accused infringer created a compatible product that did not include the copied code. *E.g. Sega*, 977 F.2d at 1527-28. Google, of course, did the opposite: It copied the code directly into a competing product and made that product *in*compatible.

B. Google next turns to factor 1, addressing only half of it. Pet. 24-28. In assessing "the purpose and character of the use," courts decide whether it is commercial and whether it is "transformative." *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 578-79 (1994). Google skips right over the holding that its copying was wholly commercial. App. 25a-28a. That weighs heavily against fair use—so heavily that it could have dominated factor 1, even if Android were modestly transformative. *See Monge v. Maya Magazines*, *Inc.*, 688 F.3d 1164, 1183 (9th Cir. 2012). As to the half that Google does address, it agrees the court stated the right test for transformative use: A use is transformative if it "alter[s] the first [work] with new expression, meaning, or message." Pet. 24 (quoting *Campbell*, 510 U.S. at 579).

Google's theory is that Android transformed Oracle's work because the Java platform was designed for personal computers and laptops, whereas Google adapted it for the "new context" of mobile devices, such as tablets and smartphones. App. 35a (emphasis omitted). The Court of Appeals rejected that argument for two independent reasons, only one of which Google challenges.

First, the uncontroverted evidence showed Google's premise was false: "Java SE APIs were in smartphones before Android entered the market." *Id.*[5] That included undisputed evidence that Oracle licensed its work to Danger for a smartphone and to SavaJe to develop its own smartphone platform. App. 50a. That meant smartphones were not a new context. That holding is dispositive on factor 1.

Google's "fundamental" complaint (Pet. 24) challenges the court's *alternative* holding: that, even if smartphones were a new context for Oracle's work, copying Oracle's code from larger computers (PCs) to smaller ones (smartphones) was not transformative because the code was used "for the same purpose" in both contexts—to enable programmers to remember, locate, and run prepackaged programs. App. 33a-35a. Google argues that the court erred in "fixating only on the material that Google reused"—the specific declaring code it copied—rather than the material Google added. Pet. 24.

Google is wrong. "[N]o plagiarist can excuse the wrong by showing how much of his work he did not pirate." *Harper & Row*, 471 U.S. at 565 (quotation marks omitted). No matter how much material one adds, it is not transformative to "merely use[] [the

---

[5] Google's smartphone/PC dichotomy is false for another reason. Just as the evidence closed, Google went public with a revelation that it had withheld in discovery: that it was about to bring Android to PCs. The district court also excluded evidence that Android and Java competed head-to-head in other device markets. These were separate bases for appeal, which the Court of Appeals did not reach. *See* App. 8a-9a, 35a n.7; *Oracle II*, Oracle Br. 55-78.

copied material] to get attention or to avoid the drudgery in working up something fresh." *Campbell*, 510 U.S. at 580. What Google added to adapt Oracle's work from PCs to smartphones is no different from what a producer does to convert a short story into a movie, which this Court has found to be the "classic … unfair use." *Stewart v. Abend,* 495 U.S. 207, 238 (1990) (quotation marks omitted).

All the courts of appeals (including the decisions Google cites) reject the approach Google advocates. They ask whether the new work "uses the *copyrighted material itself* for a purpose, or imbues it with a character, different from that for which it was created." *TCA Television Corp. v. McCollum*, 839 F.3d 168, 180 (2d Cir. 2016) (emphasis added) (using "Who's on First" routine in new context of "dark[]" play not transformative, because the lines of the routine have the same meaning), *cert. denied*, 137 S. Ct. 2175 (2017); *accord Dr. Seuss Enters., L.P v. Penguin Books USA, Inc.*, 109 F.3d 1394, 1400-01 (9th Cir. 1997) (moving memorable elements of "The Cat in the Hat" to the context of the "OJ Simpson trial" did not transform those features); s*ee generally Oracle II*, NYIPLA Br.[6]

---

[6] The cases Google cites each found that the *material copied* changed meaning in the new work. *Am. Soc'y for Testing v. Pub.Resource.Org.*, *Inc.*, 896 F.3d 437, 448 (D.C. Cir. 2018) (new work reproducing industry safety standards adopted into law gave standards new meaning through informing public of the law); *Swatch Grp. Mgmt. Servs. Ltd. v. Bloomberg L.P.*, 756 F.3d 73, 84-85 (2d Cir. 2014) (new work reporting earnings call conveyed different message); *Seltzer v. Green Day, Inc.*, 725 F.3d 1170, 1177 (9th Cir. 2013) (new work changed message

Google also complains that "[t]he Federal Circuit effectively dismissed … concerns regarding lock-in effects and interoperability" in analyzing transformative use. Pet. 27. But Google did not "rely on any interoperability arguments [on] appeal." App. 45a-46a. It "abandoned" arguments about interoperability, App. 46a n.11, and the phrases "lock-in," "barriers to entry," "anti-competitive," and their synonyms appeared nowhere in its brief. Nor did it present any such evidence at trial.

Why? Because the evidence showed that any such assertion was not only wrong but utterly hypocritical. Oracle liberally licenses its work, even to competing platform developers. *Supra* 6-8. Its non-negotiable condition is that developers must comply with the golden rule of compatibility: "write once, run anywhere." Java's raison d'etre was compatibility. Google is the only commercial platform developer ever to refuse. Worse, it copied the API packages and consciously made Android *in*compatible with Java. App. 32a, 46a n.11. It broke "write once, run anywhere." Programmers who write for Android (and their programs) are locked into Android because their programs do not run on other platforms. Google thus cannot now argue that considerations of interoperability compel a finding of fair use.

Google protests that under the Court of Appeals' approach "the reuse of any preexisting computer code in new software would never fall within the fair-use defense." Pet. 27. But the court said explicitly that it did "not conclude that a fair use defense could never

of original from "themes of youth culture" to criticism of "the hypocrisy of religion").

be sustained in an action involving the copying of computer code." App. 53a-54a. It cited prominent cases finding uses of computer code transformative. *Id.*

C. Google seeks review of the Court of Appeals' application of the substantial evidence standard of review with respect to factor 4. Pet. 28-29. But Google concedes the court "correctly stated" the standard. Pet. 22. This Court does not sift through lengthy trial records to spot-check for material disputes of fact. In any event, the Court of Appeals applied the standard faithfully.

Factor 4 accounts for two types of harm to a work: harm to markets that the work is currently in, and harm to potential markets "that creators of original works would in general develop or *license others to develop*." *Campbell*, 510 U.S. at 590, 592 (emphasis added). The Court of Appeals properly held Google inflicted both types of harm. App. 47a-53a.

With respect to current markets, Google contends that the court overrode the jury and "made its own (erroneous) determination that Java SE was in fact used in early mobile devices comparable to Android before Android's release." Pet. 28. But Google wholly ignores harm to tablets—it nowhere disputed that Amazon switched between Java SE and Android for the Amazon Kindle and used Android to negotiate steep discounts from Oracle. App. 50a-51a. Nor was there any dispute that Java SE was used in the Danger smartphone (the T-Mobile Sidekick), which Android's *founder* described as comparable to the first Android smartphones. C.A. 50,618-19, 50,887, 51,617;

App. 50a. Either one would suffice, but Google ignored (and continues to ignore) both.

Google's challenge to the court's independent finding of harm to potential markets is even more tenuous. Google asserts that the Court of Appeals erred in employing hindsight bias to find smartphones were a market that Oracle would naturally have entered. Pet. 29. Not so. The undisputed evidence the court invoked was that, before Android ever emerged, (1) Oracle *did* view specialized platforms for mobile devices as a burgeoning market for its work, *supra* 8, 29, and (2) "Oracle and Google engaged in lengthy licensing negotiations," which "demonstrates that Oracle was attempting to license its work for … smartphones," App. 51a.

III. Google's Policy Arguments Are Meritless.

A. As it did in its prior petition, Google asserts that this Court's immediate review is necessary to avoid a "devastating impact on the development of computer software." Pet. 30. If that were so, one would expect Google to muster some evidence of devastation in the four years since this Court denied certiorari—an eternity in the software industry. But Google offers only attorney conjecture—as do the usual list of amici, many of whom have long advocated weakening statutory protections for software. Google cites nothing to suggest that programmers have "abandon[ed] their traditional building-block approach to software development," that they have stopped making interoperable products, or that copyright litigation has exploded in the face of "confusion." Pet. 31.

To the contrary, software innovation has thrived since then. *See* BSA Foundation, *The Growing $1 Trillion Economic Impact of Software* (Sept. 2017), <https://tinyurl.com/y77xjgke>. And Google's sky-is-falling rhetoric is at odds with the numerous stakeholders who filed amicus briefs supporting Oracle throughout this litigation. *Supra* 2-3. In arguing otherwise, Google and its amici exaggerate the decisions' reach in multiple ways.

First, their assertions are premised largely on characterizing the Court of Appeals' opinion as inconsistent with settled law, practice, and expectations. Pet. 4, 17. What law? An unbroken line of cases has granted copyright protection to original code less creative than the Java APIs and has found copying of code to be unfair when incorporated into a competing product. *Supra* 14, 27. Neither Google nor its amici cites a single case—in any court—that has ever found it permissible to copy this much code (or this much structure and organization) "and us[e] it for the same purpose and function as the original in a competing platform." App. 53a. A claim of settled legal expectation is vacuous, no matter how many amici Google could recruit to sign briefs attesting to their unwarranted contrary expectation.

And what settled practice? In its previous petition, Google offered examples of a supposed pervasive commercial practice of freely copying software. We refuted them, demonstrating that each was an instance of either a *licensed* use (e.g., under an open source license) or material that had *no* copyright protection. *Oracle I*, Br. in Opp. 32-33. Google does not repeat them.

Second, Google and its amici misstate whom the decision affects. Pet. 29. This case is about *Google's* copying of Oracle's work. *App programmers* are unaffected. They remain free to write any program they want availing themselves of the Java APIs, for free. The decision only requires *platform developers*—usually large corporations with tremendous resources, like Google—to take a license if they want to copy Oracle's work and incorporate it into a substitute commercial product.

Third, Google and its amici exaggerate what the opinion covers. They leave the impression that the court held that anything that can be labeled an "interface" is protected and can never be copied. But here, again, they are just capitalizing on the imprecision of their own term "software interface." *Supra* 12-13, 26. There is no special law of "software interfaces." As is true with any work, whether a "software interface" has copyright protection or can fairly be copied depends on what it is and how it is used. The Court of Appeals' holding applied to Oracle's particular work—a massive software platform with an intricate architecture of thousands of programs—and Google's particular use of that work. In contrast, some interfaces are merely uncopyrightable brief strings of characters, utterly lacking in originality and expression (as in *Lexmark*). Others might be more expressive but nevertheless fair to copy for the right purposes, such as research or to create non-infringing products. For most interfaces, the question never arises because, as amici describe, strong market incentives encourage companies to permit free reuse or offer generous licensing terms to create interoperable products. That is their prerogative, but it does not wipe out the legal

rights of others who choose to commercialize their works differently.

Relatedly, Google and its amici attribute to the court a holding that anything deemed necessary for "interoperability" is protected and cannot be copied. But the court had nothing to say about interoperability because Google waived any such argument. *Supra* 31. And the court cited favorably the Ninth Circuit's *Sega* and *Sony* decisions that permit copying in some circumstances (not applicable here, *supra* 27-28) to promote interoperability. App. 53a-54a.

Fourth, Google repeats its concerns about lock-in. In any future case where an infringer presents evidence of compatibility and lock-in, it can explore how that affects fair use. Nothing the Court of Appeals said will constrain that discussion because Google had no such evidence and made no such argument below. *Supra* 31.

B. Google stresses the importance of fostering "innovation" in software development. Pet. 4. We could not agree more. The Founders, Congress, and this Court determined that "the best way to advance public welfare" is to "encourage[]" authors to engage in exactly the sort of "individual effort" Oracle undertook, by rewarding their efforts with "personal gain." *Mazer*, 347 U.S. at 219. Oracle would never have "invested as heavily in Java" if it knew its "investment … would not receive copyright protection." *Oracle I*, McNealy Br. 3; *see Oracle I*, Microsoft Br. at 3-4, 8.

Google's theory is that, having invested all those resources to create a program popular with platform developers and app programmers alike, Oracle should

37

be required to let a competitor copy its code so that it can coopt the fan base to create its own best-selling sequel. That argument would never fly with any other copyrighted work. And as Google told the Court of Appeals, "[t]here is no reason to treat software differently." *Oracle II*, Google Br. 40. "[A]llowing Google to commercially exploit Oracle's work will not advance the purposes of copyright[.]" App. 53a. The next Oracle will think twice about investing as heavily in a venture like Java if it knows that any competitor could freely copy its work to compete directly against it.

CONCLUSION

This Court should deny the petition.

Respectfully submitted,

E. Joshua Rosenkranz
Counsel of Record
ORRICK, HERRINGTON &
 SUTCLIFFE LLP
51 West 52nd Street
New York, NY 10019
(212) 506-5000
jrosenkranz@orrick.com

Date: March 27, 2019