No. 18-956

In The
Supreme Court of the United States

GOOGLE LLC,

Petitioner,

v.

ORACLE AMERICA, INC.,

Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The Federal Circuit

BRIEF OF COPYRIGHT SCHOLARS AS AMICI CURIAE IN SUPPORT OF THE PETITIONER

REBECCA TUSHNET
Counsel of Record
HARVARD LAW SCHOOL
1575 Massachusetts Ave.
Cambridge, MA 02138
(703) 593-6759
rtushnet@law.harvard.edu

Counsel for Amici Curiae

COCKLE LEGAL BRIEFS (800) 225-6964
WWW.COCKLELEGALBRIEFS.COM

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page

CASES

TABLE OF AUTHORITIES—Continued

Page

TABLE OF AUTHORITIES—Continued

TABLE OF AUTHORITIES—Continued

Page

INTEREST OF AMICI CURIAE[1]

This brief is filed on behalf of the undersigned professors identified in Appendix A.[2] Amici are scholars whose research and teaching focus is copyright law. Amici have no direct interest in the outcome of this litigation. Amici are concerned that the Court of Appeals for the Federal Circuit's decision below misapprehends the fair use doctrine, and express no opinion on the first question on which this Court granted certiorari: whether the declarations that Google used in the Android platform are protectable elements of the Java Platform Special Edition (Java SE), the work of authorship whose copyright Oracle alleged was infringed.

SUMMARY OF THE ARGUMENT

The fair use doctrine requires courts "to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster." *Campbell v. Acuff–Rose Music, Inc.*, 510 U.S. 569, 577 (1994). The Federal Circuit's rejection of a jury finding of fair use instead embraced a rigid approach that, as a matter of law, would bar any

---

[1] Pursuant to Rule 37.6, Amici affirm that no counsel for a party authored this brief in whole or in part and that no person other than Amici and their counsel made a monetary contribution to its preparation or submission. Petitioner's consent to the filing of amicus briefs is filed with the Clerk and Respondent has consented to the filing.

[2] Amici's institutional affiliations are provided only for purposes of identification.

copying of code into a new program, even for a different platform, as non-transformative and unfair. To justify its ruling, the Federal Circuit abandoned a consensus in the lower courts about the broad scope of fair use when dealing with highly functional software elements. It made key mistakes about the fair use factors and their balancing, including inflating the relevance of commerciality, applying an erroneous "no more than necessary" standard for copying, dismissing as insignificant the highly functional nature of computer programs, and conflating the market for Java SE as a whole with the market for individual declarations. Absent these legal errors, it is clear that the jury was at least reasonable in making factual findings that supported a finding of fair use.

The extent to which a new work has a new meaning, message, or purpose—transformativeness—is often and rightly prioritized in the fair use analysis. *Campbell*, 510 U.S. at 579. But what constitutes transformativeness is often contentious. Here, the new purpose of Google's new code implementing the declarations was the creation of a new computing environment in which Java programmers could readily create programs on multiple platforms, which required the use of limited portions of highly functional declarations. This type of purpose has been recognized as transformative because of its role in furthering competition and innovation. A computer interface supports the creation of other creative works, and in such situations, it is important to avoid locking in third parties to specific

platforms. *See*, *e.g.*, *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524–27 (9th Cir. 1992).

Factors two and three of the fair use test help define the boundaries of this type of fair use. In cases such as the one at bar, the highly functional nature of the copied declarations and the limited amount of the overall Java SE work used, consistent with industry practices, are vital considerations supporting the conclusion that Google's use was a transformative use that served copyright's basic goal of encouraging creation of new works. By discounting to the point of irrelevance the thinness of protection for highly functional aspects of computer interfaces (factor two) and of the industry practice of treating the amount Google reimplemented as reasonable (factor three), the Federal Circuit distorted its analysis of the other fair use factors, threatening the coherence of fair use doctrine and the ultimate progress of creativity.

Indeed, because the copyright is thin and Google copied only a very limited amount, the copying arguably failed to meet the substantial similarity standard. *See*, *e.g.*, *Experian Information Solutions, Inc. v. Nationwide Marketing Services Inc.*, 893 F.3d 1176, 1186 (9th Cir. 2018) (in dealing with thin copyrights, "infringement cannot be based on a showing that only a part of the work has been copied"); *Ross, Brovins & Oehmke, P.C. v. Lexis Nexis Group,* 463 F.3d 478, 482–84 (6th Cir. 2006) (the defendant selected a set of legal forms, 61% of which were forms included in the plaintiff's compilation; for purposes of infringing similarity, that was not particularly high given the thinness of

plaintiff's copyright); *Schoolhouse, Inc. v. Anderson*, 275 F.3d 726, 729–30 (8th Cir. 2002) (74% overlap where copyright was thin was insufficient). In the frame of fair use as well, the Court should not lose sight of the overriding importance of confining the scope of rights in highly functional works so as not to harm the overall creative ecosystem.

By downplaying the relevance of the nature of the work and the amount taken, the Federal Circuit fell into the well-known trap of circularity: reasoning that, because Oracle could have charged a license fee for this type of use if fair use were unavailable, Oracle suffered cognizable market harm. Because such claims can be made for any fair use, which by definition is not paid for, this reasoning cannot distinguish fair and unfair uses. But factors two and three can help identify when crediting such claimed market harm would be inconsistent with copyright's overall balance between past and future creators. Thus, given the limited copying of functional elements here, factors one and four also support fair use, because Google's purpose was generative of additional creativity by third parties and because Oracle's claimed market harm goes beyond the legitimate scope of its thin copyright in highly functional declarations.

A thin copyright for software, including Java SE, provides software copyright owners with meaningful protection against copying of significant amounts of expression, but meaningful protection does not require the expansive rights that the Federal Circuit granted. Providing a broad scope to highly functional elements

of software is unnecessary and dangerous to competition and innovation. Factors two and three enable fair use to implement this distinction between types of works. The Federal Circuit erred in not recognizing this interaction between the fair use factors and instead adopting a rigid rule that would preclude fair use in computer programs.

ARGUMENT

Oracle's successful effort before the Federal Circuit to cast the case as a matter of inherently wrongful copying by a competitor diverts attention away from the underlying economic principles of copyright. These principles aim to provide a reasonable degree of protection for copyrightable expression, while leaving room for others to build upon preexisting works and to develop new communication technologies.

Factor One: The Jury Could Reasonably Have Concluded that Google's Creation of a New Platform Favored Fair Use.

Factor one, the purpose of the accused use, considers different elements, including transformativeness and commerciality. This case involves the type of fair use that one prominent casebook labels "technical interchange," or enabling the creation of programming options in software—a specific kind of creativity-promoting transformativeness. *See* Julie E. Cohen et al., Copyright in a Global Information Economy 602 (4th ed. 2015).

Reuse of portions of an API for a functional purpose can be a transformative use of a work (here, Java SE) where it enables the creation of new creative works and enhances the overall creative ecosystem. Copying interfaces to enable programmers to more easily make programs for multiple platforms is a transformative purpose because it makes the system function better for people creating new things. Forcing needless variation in this context does not enhance creativity; it balkanizes groups of creators and wastes effort. By contrast, the jury reasonably could have found that better cross-platform compatibility allows programmers to transport their own code to different implementations; the declarations form building blocks through which larger structures can be created. The Federal Circuit thus failed to appreciate the relevance of factors two and three—the highly functional nature of the limited number of copied declarations—to a proper analysis of transformativeness.

a. Promoting Compatibility is Transformative Because It Promotes Creativity and Innovation in the Larger Software System.

1. The Federal Circuit created a rigid rule that would preclude fair use in code.

The Federal Circuit reasoned that factor one weighed against Google because the intrinsic purpose of the parties' declarations was the same. Pet. App. 31a-33a. This "intrinsic purpose" reasoning creates a per se rule against copying any interface elements or code in a computer program. It fails to weigh the

benefits of standardization for third parties, and fails to appreciate the interaction of factors two and three with transformativeness. The Federal Circuit's simplistic rationale suggests that a law review article that quoted another law review article for support would infringe the earlier article, because they both had the same "intrinsic purpose." Without assessing the role of the copied material in the parties' works, this conclusion is erroneous.

2. Android promoted compatibility for the Java community, a value distinct from "interoperability."

The Federal Circuit accepted Oracle's argument that Google's use was not transformative because it didn't achieve complete "interoperability." That is, any given Java program could not automatically run on both mobile and desktop without any changes. Pet. App. 46a & n.11. But using exact interchangeability to define the only legitimate purpose for copying declarations or code precludes anyone from using limited portions of functional libraries without Oracle's permission—they must license the whole or take nothing. Oracle's position would allow it to leverage its rights over copyrightable aspects of Java SE to forbid legitimate uses of the uncopyrightable aspects. Cf. *Omega S.A. v. Costco Wholesale Corporation*, 776 F.3d 692, 693–94 (9th Cir. 2015) (using copyright in small image to control uncopyrightable watches was misuse of copyright); *Alcatel USA, Inc. v. DGI Technologies, Inc.*, 166 F.3d 772, 793–94 (5th Cir. 1999) (using copyright to control

uncopyrightable microprocessors was misuse); *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970, 978 (4th Cir. 1990) (leveraging copyright to "suppress any attempt by the licensee to independently implement the idea which [software] expresses" was misuse); *Qad. Inc. v. ALN Associates, Inc.*, 770 F. Supp. 1261, 1269–70 (N.D. Ill. 1991) (using copyright to control functional and unowned elements of software was misuse).

As other courts have recognized, transformative compatibility does not require identicality of programs. The details of the technology and the practices of users weigh heavily in determining whether the use of aspects of an interface supports new creativity. *Sony Computer Entm't Inc. v. Connectix Corp.*, 203 F.3d 596, 607, 609 (9th Cir. 2000) (finding that the goal of compatibility favored fair use; noting that the new console did not play games as well as the original);[3] cf. *Compaq Computer Corp. v. Procom Tech., Inc.*, 908 F. Supp. 1409, 1419, 1427 (S.D. Tex. 1995) (where defendant used plaintiff's software parameters but chose its own values for them, producing different outcomes in some cases, the products were "compatible" for purposes of false advertising law; "compatibility" is a matter of practical function). Multiple witnesses testified that creating a Java programming environment on mobile devices did not require copying of Java SE as a whole or even of the APIs as a whole. Indeed, constraints such

[3] *See* Br. of Connectix Corp., *Sony Computer Entertainment Corp. v. Connectix Corp.*, No. 99-15852, at 18 (9th Cir. filed May 22, 1999) (Connectix implemented only 137 of the 242 functions implemented in Sony's BIOS).

as battery life and the way that multiple applications would run on a device required avoiding some libraries. JA49-50 (some packages were not appropriate for mobile because of battery and memory limitations, among other things); JA264 (wholesale copying of Java SE would have been inappropriate for mobile); JA168-70 (offering examples). Google used the libraries that developers would particularly expect and that would be useful on a mobile platform. JA48-49; JA203; JA213-14.

The jury could reasonably have concluded that what Google created served the standards and needs of the industry—the very reason why interoperability has value. As Sun's then-CEO announced to the public after Google launched Android, "Google . . . just strapped another set of rockets to the [Java] community's momentum"; he, like others in the industry, considered it to be a Java platform congenial to Java developers, TX2352, and believed Android would improve the overall health of Java by empowering developers, JA133. In practical terms, the quick success of Android demonstrated that Oracle's all-or-nothing "interoperability" argument did not reflect the perspective of Java programmers, who are the ones who decide where their own efforts in creating new programs should be directed.

As a result, the jury could reasonably have agreed with Google that what is important from the perspective of empowering future creators is sufficient standardization, much as the Federal Rules of Civil Procedure standardize key elements of federal practice

despite the existence of local rules. Disallowing programmers from using core elements to which they have become accustomed makes it much more difficult for them to work together (or to compete against each other on the merits of their specific contributions) to create or adapt innovative apps that can run on both Android and Java technologies. Importantly, this benefit to the programming community was achieved even though there are API libraries that are unique to particular environments, such as mobile versus desktop, and thus not used across all Java programs.

3. Creating a new platform is a purpose favoring fair use because it creates new works and also enables creation and dissemination of additional works.

The jury could reasonably have found Google's reimplementation of Java for Android to be a significant innovation that benefited the larger programming community. Prior cases have recognized the generative function of access to platforms. In *Sony Computer Entertainment v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000), for example, Connectix reimplemented the functionality of Sony's interface in its "PlayStation emulator" software. This reimplementation was correctly held to be transformative for at least two reasons. First, Connectix's emulator software enabled consumers to use legitimate, purchased copies of PlayStation games in a new environment (on personal computers). Id. at 606–07. Second, the court considered Connectix's own creation, the emulator software, "a wholly new

product, notwithstanding the similarity of uses and functions" between the PlayStation gaming console and the emulator program. This was at least modestly transformative because Connectix had created its own code for implementing the PlayStation firmware's functions in the emulator software, furthering the basic purposes of copyright. Id.

Importantly for purposes of the transformative use inquiry, both Connectix and Google reimplemented the functionalities of existing software in new computing environments and wrote their own software code to implement the relevant interface. Likewise, *Sega*, although it was decided before *Campbell*, recognized the benefit to creators in general if unlicensed parties can reimplement interfaces, consistent with *Campbell*'s recognition of the role of fair use in spurring new creativity. 977 F.2d at 1522–23 (copying highly functional elements in order to create a new computing environment was "a legitimate, essentially non-exploitative purpose" which "led to an increase in the number of independently designed video game programs offered for use with the Genesis console. It is precisely this growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote"); *see also Lotus Development Corp. v. Borland Intern., Inc.*, 49 F.3d 807, 821 (1st Cir. 1995) (Boudin, J., concurring) (suggesting that a commercial use should not infringe where third parties have invested in their own efforts that have increased value when they are compatible

across different implementations of software). Although both *Sony* and *Sega*, unlike the Federal Circuit below, concluded that interfaces (the BIOS in *Sony* and interface procedures in *Sega*) were uncopyrightable, their reasoning on the creativity-promoting transformativeness of the resulting overall software applies with equal force even assuming that what is incorporated in the new work is copyrightable.

b. The Federal Circuit inappropriately weighed commerciality against Google.

Precisely because so many favored fair uses are for profit, the Court wisely established that the fact that the use is made by a for-profit endeavor has little weight against fair use as long as other elements of the test favor fairness. *Campbell*, 510 U.S. at 579, 584.

The Federal Circuit went beyond a definition of commerciality as for-profit use and reasoned that, although Google did not sell its Java implementation, giving customers for free something they would ordinarily have to buy can constitute commercial use. Pet. App. 27a (quoting *A&M Records, Inc. v. Napster, Inc.*, 239 F.3d 1004, 1015 (9th Cir. 2001)). As the Federal Circuit used this formulation, it is mistaken. First, it is factually erroneous: customers never have to buy Java—Oracle demanded that Google, but not ordinary programmers, purchase a license.

Second and more significantly, the framing of commerciality as a matter of whether payment is ordinary is both inconsistent with the plain meaning of

"commercial"[4] and mistakenly duplicates factor four (the effect on the market for the original work). *See Sony*, 464 U.S. at 450 n.33 (rejecting the argument that a use is necessarily commercial if it replaces a sale); Mark Bartholomew & John Tehranian, An Intersystemic View of Intellectual Property and Free Speech, 81 George Washington Law Review 1, 22 (2013) ("Courts have stretched the definition of commercial use to include activities that ordinarily would not be viewed as commercial in any other context, especially when First Amendment rights are at issue. . . . [Market harm] concerns are already covered by the fourth factor of the test and need not be redundantly considered in the first."); Matthew Sag, Predicting Fair Use, 73 Ohio St. L.J. 47, 60 (2012) ("[T]his construction of commerciality is almost exactly equivalent to the concept of market effect under the fourth factor, so employing it here would be double counting to say the least.").[5]

---

[4] Webster's relevantly defined the adjective "commercial" as "engaged in work designed for the market," "of or relating to commerce," "characteristic of commerce," "viewed with regard to profit," and "designed for a large market," with "commerce" relevantly defined as "the exchange or buying and selling of commodities on a large scale. . . ." Webster's New Collegiate Dictionary 226 (1973).

[5] The Federal Circuit's statement that "[d]irect economic benefit is not required to demonstrate a commercial use," 886 F.3d at 1198 (quoting *A&M*, 239 F.3d at 1015), is also suspect, as the Federal Circuit applied it. "[N]onprofit educational uses," 17 U.S.C. §107(1), may contribute to the quality of education, leading to additional donations or resources for the institution; successful criticism may enhance a scholar's career. Given the

c. Good or bad faith are generally unhelpful terms in fair use, but, to the extent it is considered, Google acted in good faith.

In its arguments below, Oracle argued that Google's unsuccessful pursuit of a license for additional rights to trademarks and copyrights evinced bad faith, which should weigh against Google in the factor one analysis. Pet. App. 95a. Oracle's reasoning flies in the face of this Court's sound precedent. *Campbell*, 510 U.S. at 585 n.18 ("2 Live Crew's actions [in unsuccessfully seeking permission] do not necessarily suggest that they believed their version was not fair use; the offer may simply have been made in a good-faith effort to avoid this litigation. If the use is otherwise fair, then no permission need be sought or granted.") (citing *Fisher v. Dees*, 794 F.2d 432, 437 (9th Cir. 1986) ("[T]o consider Dees blameworthy because he asked permission would penalize him for this modest show of consideration.")).

As the district court noted, the license about which the parties had been negotiating was for a wider use of the full range of the Java technologies (as well as for co-branding). Pet. App. 107a. Even if such extensive uses required a license, Google could reasonably believe that using only some of the Java declarations and classes was lawful.

---

breadth with which the Federal Circuit, and some other courts, have applied the concept of indirect benefit, no fair use could escape condemnation for "commerciality," demonstrating this formulation's lack of utility as a consideration in a balancing test.

More generally, amici believe that "good faith" is often an unhelpful framing in factor one. *See Campbell*, 510 U.S. at 585 n.18; Hon. Pierre N. Leval, Toward a Fair Use Standard, 103 Harv. L. Rev. 1105, 1128 (1990) ("Whether the secondary use is within the protection of the [fair use] doctrine depends on factors pertinent to the objectives of the copyright law and not on the morality or motives of either the secondary user or the copyright-owning plaintiff."). It has little relationship to the statute's language or purpose:

> [F]rom the standpoint of faithfulness to statutory language, a user's course of dealing with the holder of copyright in the underlying work has little relation to the 'purpose' of the use. Second, and more important, there is little reason to infuse the doctrine of fair use with notions of commercial ethics. Unlike the doctrine of trade secrecy, the doctrine of fair use has no substantial basis in commercial morality. Like copyright law generally, fair use has an economic purpose.

Jay Dratler, Jr., Distilling the Witches' Brew of Fair Use in Copyright Law, 43 U. Miami L. Rev. 233, 334 (1988) (footnotes omitted); *see also* Simon J. Frankel & Matt Kellogg, Bad Faith and Fair Use, 60 J. Copyright Soc'y U.S. 1, 36 (2012) (reaching the same conclusions).[6] However, as discussed below, in cases

---

[6] The Court in *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539 (1985), characterized good faith as part of fair use. There, however, the defendant preempted the copyright owner's well-recognized right of first publication, which it had

involving only limited copying, a justified belief that the copying was appropriately limited bears on the reasonableness of the amount taken and thus on the third factor.

To the extent that good or bad faith matters independently, the Federal Circuit's treatment was illogical: it stated that only bad faith was relevant, and that evidence of good faith could only be used to rebut bad faith. Pet. App. 39a. The jury could legitimately take into account Google's reasonable belief in the rightness of its acts to support fair use. *See*, *e.g.*, *Bouchat v. Baltimore Ravens Ltd. Partnership*, 737 F.3d 932, 942 (4th Cir. 2013); *Gulfstream Aerospace Corp. v. Camp Sys. Int'l, Inc.*, 428 F. Supp. 2d 1369, 1378 n.5 (S.D. Ga. 2006).

Among other things, the American Committee for Interoperable Systems (ACIS), of which both Sun Microsystems and Oracle were members, endorsed and helped entrench the general understanding that reimplementing APIs was not infringing. ACIS said: "The rules or specifications according to which data must be organized in order to communicate with another program or computer, i.e., interfaces and access protocols, are not protectable expression under copyright law." *See* American Committee for Interoperable Systems (ACIS), Statement of Principles, available at https://www.ccianet.org/wp-content/uploads/2014/10/ACIS-Letter-to-Clinton-Admin-1992.pdf. Sun was a founding member

---

planned to exploit. Bad and good faith are neither necessary for such a result nor helpful outside that context.

of ACIS, and Oracle joined ACIS in 1992. Brief Amicus Curiae of American Committee for *Interoperable Systems at iii, Eng'g Dynamics, Inc. v. Structural Software, Inc.*, 46 F.3d 408 (5th Cir. 1995) (No. 92-3444) (listing ACIS members). ACIS also told this Court:

> Unlike traditional literary works such as novels and plays that stand alone and do not need to interact with any other work, computer programs never function alone; they function only by interacting with the computer environment in which their developers place them. . . . As a consequence, no matter how much better or cheaper the new program is, it will not enjoy a single sale if it cannot interoperate in its intended environment. If the developer of one part of the environment can use copyright law to prevent other developers from writing programs that conform to the system of rules governing interaction within the environment—interface specifications, in computer parlance—the first developer could gain a patent-like monopoly over the system without ever subjecting it to the rigorous scrutiny of a patent examination.

Brief Amici Curiae of American Committee for Interoperable Systems and Computer & Communications Industry Ass'n in Support of Respondent, *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 516 U.S. 233 (1996) (No. 94-2003), 1995 WL 728487 at *4-5.

Factor Two: The Federal Circuit Erred by Discounting the Highly Functional Nature of the Work and the Thinness of the Copyright

The "nature" of the work includes both the quantity of originality and the role of that originality in how the work is used or received by the public. The jury heard evidence that the declarations and classes of the Java SE API were functional, not merely in the way that all computer code performs a function, but specifically in that these particular declarations perform their mini-duties in noncreative ways. Pet. App. 113a-14a. The Federal Circuit stated that the jury could reasonably have found that the Java API as a whole merited only "thin" copyright protection. Pet. App. 42a.[7]

The Federal Circuit correctly labeled the nature of the work as favoring Google, but it failed to put this factor in proper context and thus to give it actual weight, instead reasoning that the second factor did not matter to fair use determinations generally. Pet. App. 42a-43a. Although the fictional nature of a work may not weigh *against* fair use when an accused use is transformative in a way that creates a new, critical meaning or message, *Campbell*, 510 U.S. at 586, that does not mean that the second factor should be discounted in every case. There are four factors because various fair uses can be fair for different reasons, and the factors may thus differ in relevance across types of cases. Unsurprisingly, none of the cases cited by the

[7] As stated above, Amici take no position on the copyrightability of the API as such.

Federal Circuit in support of always discounting the second factor dealt with functional aspects of computer programs. Cases that have considered "technical interchange" situations, such as *Connectix*, 203 F.3d at 603, have wisely placed special emphasis on the highly functional features of interfaces.

Here, the reasons that factor two favored Google are highly relevant to the overall fair use analysis. The Ninth Circuit has observed:

> The second statutory [fair use] factor . . . reflects the fact that not all copyrighted works are entitled to the same level of protection. . . . Works of fiction receive greater protection than works that have strong factual elements, such as historical or biographical works, or works that have strong functional elements, such as accounting textbooks.

Sega, 977 F.2d at 1524 (citations omitted). Copyright protection "does not extend to the ideas underlying a work or to the functional or factual aspects of the work." Id. As a result, even if copyright protects a highly functional work as a whole, "[u]nder the Copyright Act, if a work is largely functional, it receives only weak protection." Id. at 1527. This result, said the court, was "neither unfair nor unfortunate. It is the means by which copyright advances the progress of science and art." Id. (quoting *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 350 (1991)).

The functional character of computer programs means that they "hover" near "the elusive boundary line described in [17 U.S.C.] §102(b)." *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693, 704 (2d Cir. 1992). Owing to software's hybrid role as both "literary expression" and a "highly functional, utilitarian component in the larger process of computing," copyright provides only a "weak barrier" of protection; this narrow scope "flows from applying, in accordance with Congressional intent, long-standing principles of copyright law." Id. at 712. No matter how much work and how many choices went into producing Java SE, the highly utilitarian nature of declarations means that copyright grants them thin scope at best.

This thinner scope of protection naturally leads to a broader scope for fair use, especially in conjunction with factor three. *Connectix*, 203 F.3d at 603, 605 (fair use was an appropriate way to "preserve[] public access to the ideas and functional elements embodied in copyrighted computer software," and factor two "strongly favored" fair use); *Sega*, 977 F.2d at 1522, 1526 (same). The functional nature of software favors fair use because of the desirability of enabling second-comers to build on the functional elements of existing programs in creating new works of authorship. In the software context, then, finding fair use when there is partial rather than wholesale copying is particularly likely to serve copyright's purpose of encouraging new creativity.

Factor Three: The Reasonable Amount Taken from the Accusing Work as a Whole, Java SE

Google took only a small amount—the amount it deemed appropriate for Java programmers to be able to program for Android—which strongly favored fair use. As this Court has explained, the extent of permissible copying varies with the purpose and character of the use. *Campbell*, 510 U.S. at 586–87. The question is not whether, in judicial hindsight, the defendant took more than absolutely necessary, but whether the amount taken was "reasonable in relation to the purpose of the copying." Id. at 586. The Federal Circuit disregarded this governing standard by applying a far more restrictive and unpredictable rule: considering the amount taken to be at best neutral and "arguably" to weigh against Google, Pet. App. 47a, because it was "more than necessary" to write in Java, Pet. App. 45a.

The Federal Circuit's rule also ignored the context of software interfaces. Because any copyright in highly functional elements is necessarily thin, limited copying should weigh significantly in favor of fair use. Without such a rule, thin copyrights will expand past their statutory boundaries. Moreover, as the jury heard and evidently credited, Google took an amount from Java SE considered in the industry to be reasonable in light of its purpose of developing new, compatible works. Given the limited copying, this factor strongly favored Google. *Sega*, 977 F.2d at 1526–27.

1. Google took a small part of Java SE, the relevant work.

One aspect of the Federal Circuit's error was its emphasis on the 37 packages that Google copied in "entirety." Pet. App. 45a. In a fair use analysis, it is vitally important to identify the allegedly infringed work or works. As the statute commands, the proper inquiry considers the amount taken "in relation to the copyrighted work as a whole." 17 U.S.C. §107. Without careful attention to the boundaries of works, especially unfamiliar works such as software, plaintiffs can manipulate their claims to artificially increase the relative size of what was taken. Justin Hughes, Size Matters (Or Should) in Copyright Law, 74 Fordham L. Rev. 575, 579–80 (2005) (explaining dangers of recognizing "microworks" in software and elsewhere); id. at 613 ("If our goal is to create special incentives for the building of houses, we do not necessarily need special incentives for the making of bricks or the mixing of mortar. . . . Without such evidence that our bricks—short phrases, titles, evaluations—are under baked, so to speak, we should prevent the law from moving in that direction. . . ."); Margot E. Kaminski & Guy A. Rub, Copyright's Framing Problem, 64 UCLA L. Rev. 1102, 1118–19 (2017) (explaining that the Federal Circuit's earlier decision "did not explain why the quantitative component of its inquiry should zoom in on this one small function rather than conceive of it as a subpart of the much larger software"); id. at 1142–44 (explaining the interaction between work size and fair

use; noting the risks of a plaintiff "gaming" the work's size).

Here, the registrations as well as industry practice make clear that the work at issue is Java SE.[8] All of Java SE had about 5 million lines of code, including aspects such as the Java Virtual Machine and a Java compiler. JA212.[9] Google's implementation expressing the declarations at issue (the ones that Google reimplemented in its own code from the 37 packages at issue) totals about 11,500 lines, out of what was, at the 2008 release of Android, up to 15 million lines of code. Id. Even within the subcategory of the 37 packages, Google's own implementations accounted for 97% of the code. Pet. App. 219a. Google needed only those parts of the Java SE API that were appropriate for the smartphone context (as compared with the laptop and server systems for which Java was originally designed). As a quantitative matter, what was copied was a rounding error—for both Android and Java SE. Yet the Federal Circuit characterized Google as "taking a copyrighted work verbatim," Pet. App. 53a, as if the fractional part were the whole.

---

[8] Oracle's registrations submitted to the court were for Java SE 1.4 (for "new and revised computer code and accompanying documentation and manuals," TX464), and the same for Java SE 5.0, TX475, along with a supplemental certification of registration with alternate titles for Java SE 5.0, TX476.

[9] The part of Java SE that was the libraries (the full 166 packages) had about 2.86 million lines of code. JA212.

2. Copying highly functional elements weighs strongly in favor of fair use.

The jury reasonably could have found that the qualitative character of the copying—its limitation to highly functional declarations—likewise militates strongly in favor of fair use. The jury reasonably could have accepted that Google was indifferent to the expression of Java SE and sought only the function. *See* Wendy J. Gordon, How Oracle Erred: The "Use/Explanation Distinction" and the Future of Computer Copyright, in Copyright in an Age of Limitations and Exceptions 319, 320 (Ruth Okediji, ed., 2017) ("[A] copyright owner should have no prima facie rights over copying behavior where (1) the goals of the copying are 'use' (behavior in the realm of utility patent) and (2) the copying is done solely for goals unrelated to the expressiveness of the plaintiff's work of authorship. . . . Interoperability is one of the few areas where indifference to expression is clear: After all, when one wants a spare key made, the elegance or beauty of the key's shape is irrelevant—all that matters is that the shape fits the lock."). Cf. *Lexmark Intern., Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 544 (6th Cir. 2004) (if a programmer designed a program that would only work with the input of a copyrightable poem as the key, then copyright imposes no barrier to use of the key, even if the poem would be protected outside that context).

Along with ignoring the highly functional nature of what was taken in its qualitative evaluation, the Federal Circuit engaged in erroneously circular reasoning.

By reasoning that, because copying benefited Google in making Android, there was qualitative significance, Pet. App. 46a, the Federal Circuit remade factor three so that it would always weigh against fair use. *See* 1 Melville B. Nimmer & David Nimmer, Nimmer on Copyright §13.05[A][3], at 13–180 (2000) ("To avoid circular reasoning, the plaintiff manifestly should not be heard to argue that the defendant's copying of brief passages vouchsafes their qualitative significance." (footnote omitted)).

As the district court detailed, the jury could have found that Google used highly limited portions of Java SE's APIs as part of an innovative mobile device platform. Pet. App. 117a. Google not only wrote its own implementing code for the portions of Java SE that it used, but also created many new declarations to enable a vast array of additional, innovative smartphone functionalities. Id. at 105a. Furthermore, the portions of Java SE that Google reimplemented may have helped preserve consistency of use within the larger Java developer community. Id. at 104a-05a. Given this, the jury could reasonably have concluded that Google's use of limited portions of Java SE was appropriate to achieve its purpose.

Oracle also argued that Google copied the structure, sequence and organization (SSO) of the Java SE API libraries. Because of the ways in which declarations are named, that SSO necessarily came with the names, but the jury could reasonably have found that this was not relevantly "additional" copying of creative expression, any more than the pagination of a print

volume of caselaw that naturally follows from it having cases in some order adds additional expression. *See Matthew Bender & Co. v. West Pub'g Co.*, 158 F.3d 693, 699 (2d Cir. 1998). If it is fair to use the individual declarations, then it is fair to use the SSO that necessarily comes along with them.

Testimony in the fair use trial record, including by Sun's last CEO,[10] supported Google's contention that there was a common understanding in the software industry that programmers were free to reimplement declarations in independently written code.[11] This reasonable belief bears on the appropriateness of the quantitative and qualitative amount taken by Google in writing its own code. As the District Court noted, Google's witnesses

> testified that they had understood that "reimplementing" an API library was a legitimate, recognized practice so long as all that was duplicated was the "declaring code" and so long as the duplicator supplied its own "implementing code," that is, the methods were

---

[10] JA113-15, 119-20 (testifying to an understanding from the outset of Java's release that implementations were protected but API itself was not).

[11] *See*, *e.g.*, JA246 (testifying to "established industry practice" that the API could be reimplemented without permission); JA164-65 (same); JA153, 157 (same; witness was employed at Sun during the relevant period); JA126-28, 55 (openness of the API "was a way to try to make things fair"; that free use of APIs, even by competitors, was acceptable and promoted accessibility of Java to programmers; that APIs were "open" and "then we compete on implementations" without being "locked into one company's implementation").

> "re-implemented." In this way, Java programmers using the Android API could call on functionalities with the same Java command statements needed to call the same functionalities in the Java API, thereby avoiding splintering of the ways that identical functionalities became invoked by Java programmers.

Pet. App. 96a. The distinction between interfaces and implementations is long-standing in the computing field. *See*, *e.g.*, Alfred Z. Spector, Software, Interface, and Implementation, 30 Jurimetrics J. 79, 85–86 (1989). The industry consensus was strengthened by several appellate court cases, including two significant Ninth Circuit precedents, that rejected copyright claims involving computer program interfaces.[12]

---

[12] *See*, *e.g.*, *Sega*, 977 F.2d at 1514–15, 1522 (characterizing the Sega interface reimplemented by unauthorized game developer as unprotectable "functional requirements for compatibility"); *Connectix*, 203 F.3d at 602–03 (software that emulated a game platform's functionality by reimplementing its BIOS interface in independently written code, allowing owners of PlayStation games to play on unlicensed consoles, used only unprotectable material); *see also Altai*, 982 F.2d at 714–15 (rejecting infringement claim based on copying the structure, sequence, and organization of plaintiff's list of services as well as parameter lists that set forth interfaces for interacting with third-party programs because the list of services "was dictated by the nature of other programs with which it was designed to interact" and the parameter lists were not similar enough to infringe); *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532, 1539, 1543–46 (11th Cir. 1996) (holding that jury must be allowed to consider evidence and argument that some literal copying of code was necessary so that its new operating system could execute its own application program); *Lotus*, 49

Indeed, a large number of books have been published that set forth the Java SE API, in whole or in part, including the declarations and class structures. *See*, *e.g.*, Ian F. Darwin, Java Cookbook: Solutions and Examples for Java Developers (3d ed. 2014). These books aim to explain how to use the Java API to develop new software programs. These for-profit publications often reproduce the whole or substantial parts of the Java API. The District Court took note of these books in explaining its denial of Oracle's Rule 50 motion:

> Many thousands of pre-written methods have been written for Java, so many that thick books . . . are needed to explain them, organized by packages, classes, and methods. For each method, the book sets forth the precise declaring code but does not (and need not) set forth any implementing code. In other words, the book duplicates all of the method declarations (organized by packages and classes) together with plain English explanations. A Java user can study the book and learn the exact method name and inputs needed to invoke a method for use in his or her own program. . . .

Pet. App. 101a-02a. Where others were freely copying large parts or the entirety of the API, the jury could

---

F.3d at 815–18 (holding that the selection and arrangement of specific commands for invoking specific functions of Lotus's popular 1-2-3 spreadsheet program was unprotectable).

have found that Google's far more limited copying strongly favored Google.

In this way, the thinness of the copyright—factor two—interacts with factor three: even if APIs as a whole cross the line into copyrightability, Google should be able to use declarations (and the SSO they necessarily reflected) that were reasonably necessary to pursue its legitimate goal of enabling the creation of an environment accessible to Java programmers.

The Federal Circuit dismissed Google's reasons for copying by claiming that "there is no inherent right to copy in order to capitalize on the popularity of the copyrighted work or to meet the expectations of intended customers." Pet. App. 46a. Not only is this reasoning wrongly indifferent to the limited qualitative and quantitative amount copied, it disregards the importance of competition and third-party creation. The work at issue is not a popular film that can stand on its own, but a platform for creating new programs. While there is no right to copy wholesale simply because a work has achieved marketplace success, its dominance can justify taking reasonable amounts in order to create a new platform that enhances the capabilities of third parties to create new works.

Given the testimony about the common understanding about freedom to reimplement APIs, bolstered by precedent and by commercial copying of the Java API in books, the amount taken was appropriately limited based on Google's purpose and strongly favored fair use.

Factor Four: Circularity and the Federal Circuit's Conflation of the Market for Java SE with the Hypothetical Market for Declarations

The Federal Circuit's analysis of factor four was entirely and improperly circular: If what Google did was fair use, then Oracle would lose a potential license fee. But that is always true when fair use is found. Pet. App. 268a ("[F]air use asks whether the accused infringer needed a license for its use in the first place. Thus, consideration of harm to the potential market to license works like accused infringer's begs the question of fair use.").

In some situations involving traditional creative works with "thick" copyrights, experience and normative commitments to free speech break the circle: the markets for criticism, educational uses, and parody are not legitimate even if a specific copyright owner evinces a willingness to license particular instances. *Campbell*, 510 U.S. at 592 ("[T]here is no protectible derivative market for criticism."); *Castle Rock Entertainment Inc. v. Carol Publishing Group*, 150 F.3d 132, 146 n.11 (2d Cir. 1998) (holding that a copyright owner cannot control fair use markets merely "by developing or licensing a market for parody, news reporting, educational or other transformative uses of its own creative work"); *see also Cambridge Univ. Press v. Patton*, 769 F.3d 1232, 1278 (11th Cir. 2014) ("Plaintiffs may not head off a defense of fair use by complaining that every potential licensing opportunity represents a potential market for purposes of the fourth fair use factor.").

When it comes to software, the benefits of interoperability and allowing third parties to avoid lock-in to particular platforms similarly explain why licensing interface reimplementation in newly written code is likewise not a legitimate copyright market. In *Sega*, for example, blocking access to a larger market was not a legitimate economic interest under factor four, even though doing so would benefit Sega economically and even though Sega had established a licensing market for its interface. 977 F.2d at 1523–24.

Here, as in *Sega* and *Connectix*, factors two and three can be used to avoid the vice of circularity. *See* Hughes, supra, at 619 (discussing interaction between small amount of copying and market effect). The jury could reasonably have found that there is a fundamental difference between, on the one hand, full copies of Java SE or revisions that alter Java SE as a whole sufficiently to constitute a new derivative work, and, on the other, limited copying of highly functional aspects of Java SE.

In reversing the jury's verdict, the Federal Circuit wrongly relied on the circular claim that Google could have taken a license for the precise use it made and that this possibility established market harm. Pet. App. 51a. But the evidence did not establish harm to the market for "the work," §107(4), or even the existence of a market for a limited number of highly functional declarations. As Larry Page testified, the parties' licensing negotiations were for far broader uses of Java SE, and included proposed trademark/brand licensing. JA245. The Federal Circuit, in substituting its view of the evidence for the jury's, ignored the obvious

differences between the market for licensing all aspects of Java SE and the market for a small number of declarations within specific libraries. Those two things are not the same, any more than the market for an entire book is the same as the market for short critical or illustrative excerpts. This distinction is especially salient because, to the extent that Oracle seeks the right to license functional aspects of its APIs, it attempts to extend its rights beyond that which copyright allows; this cannot be considered a traditional, reasonable, or likely to be developed market for the rights conferred by copyright. *Authors Guild v. Google, Inc.*, 804 F.3d 202, 224 (2d Cir. 2015) (cognizable market effect must be based on copyrightable aspects of what was copied, not on uncopyrightable aspects); *Connectix*, 203 F.3d at 607–08 (competition is not market harm); *Sega*, 977 F.2d at 1523–24 (same).

Thus, the jury's fair use verdict did not undermine Oracle's derivative work right; by its plain terms, Section 107 limits all of the copyright owner's exclusive rights, including the derivative work right. 17 U.S.C. §107; *Authors Guild, Inc. v. HathiTrust*, 755 F.3d 87, 95 (2d Cir. 2014). Oracle argued that Android harmed the market for Java ME, its own mobile product, but the jury heard testimony from Sun's former head of ME licensing that Java ME was created *before* Java SE and contained no identifiable expression from the work in suit, Tr. 1668:13-19, Tr. 1691:22-1692:11; competition with a different product is not cognizable market harm.

The derivative work right has an important role to play, but that role is not to shrink fair use. *See* Pamela Samuelson, The Quest for a Sound Conception of Copyright's Derivative Work Right, 101 Geo. L.J. 1505, 1538 (2013). When what is taken from the copyright claimant's work is subject to "thin" protection at best, as discussed under Factor Two, fair use assures that the derivative work right does not vitiate the well-justified limits on the scope of a copyright owner's rights in a highly factual or functional work.

CONCLUSION

Because the Federal Circuit ignored the special characteristics of software interfaces and applied the fair use factors rigidly, its fair use analysis was distorted. It gave highly functional software protection far beyond what is necessary to prevent piracy, to the detriment of competition and creativity generally. Its fair use holding should be reversed.

Respectfully submitted,

REBECCA TUSHNET
Counsel of Record
HARVARD LAW SCHOOL
1575 Massachusetts Ave.
Cambridge, MA 02138
(703) 593-6759
rtushnet@law.harvard.edu

Counsel for Amici Curiae