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 FOR THE AMERICAN ANTITRUST INSTITUTE AS AMICUS CURIAE IN SUPPORT OF PETITIONER

RANDY M. STUTZ
Counsel of Record
AMERICAN ANTITRUST INSTITUTE
1025 Connecticut Ave., NW
Suite 1000
Washington, DC 20036
(202) 905-5420
rstutz@antitrustinstitute.org

January 13, 2020

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES ii

INTEREST OF AMICUS CURIAE 1

INTRODUCTION 2

SUMMARY OF ARGUMENT 4

ARGUMENT 7

I. AFFORDING COPYRIGHT PROTECTION TO SOFTWARE INTERFACES CAN ENTRENCH DOMINANT FIRMS PROTECTED BY NETWORK EFFECTS 7

A. Copyrights on Software Interfaces Risk Lock-in and Holdup 7

B. Lock-In Can Cement Software-Based Monopolies 9

II. COPYRIGHT'S LIMITING DOCTRINES ARE DESIGNED TO PRESERVE INNOVATORS' ABILITY TO CREATE INTEROPERABLE AND COMPETITIVE PRODUCTS AND SERVICES 12

A. Section 102(b) 13

B. Merger 16

C. Fair Use 17

III. THE COURT SHOULD SANCTION UTILITARIAN TRANSFORMATIONS OF SOFTWARE INTERFACES AS FAIR USE 19

CONCLUSION 23

TABLE OF AUTHORITIES

Page

CASES

A.V. ex rel. Vanderhye v. iParadigms, LLC,
562 F.3d 630 (4th Cir. 2009) 20

Apple Computer, Inc. v. Formula Int'l, Inc.,
725 F.2d 521 (9th Cir. 1984) 16

Apple, Inc. v. Motorola, Inc.,
757 F.3d 1286 (Fed. Cir. 2014) 7

Authors Guild v. Google, Inc.,
804 F.3d 202 (2d Cir. 2015) 6, 20

Bikram's Yoga College of India v. Evolation Yoga, LLC, 803 F.3d 1032 (9th Cir. 2015)..... 14

Bonito Boats, Inc. v. Thunder Craft Boats, Inc.,
489 U.S. 141 (1989) 12

Campbell v. Acuff-Rose Music, Inc.,
510 U.S. 569 (1994) 19

Commonwealth Sci. & Indus. Research Org. v. Cisco Sys., Inc., 809 F.3d 1295 (Fed. Cir. 2015)..... 7

Computer Associates International, Inc. v. Altai,
982 F.2d 693 (2d Cir.1992) 14, 16

eBay Inc. v. MercExchange, L.L.C.,
547 U.S. 388 (2006) 7

Fox News Network, LLC v. Tveyes, Inc.,
883 F.3d 169 (2d Cir.) 20

Harper & Row Publishers, Inc. v. Nation Enters.,
471 U.S. 539 (1985) 22

Lotus Dev. Corp. v. Borland Int'l, Inc.,
49 F.3d 807 (1st Cir. 1995) .....*passim*

Oracle Am., Inc. v. Google Inc. (*Oracle I*),
750 F.3d 1339 (Fed. Cir. 2014) .....*passim*

Oracle Am., Inc. v. Google LLC (*Oracle II*),
886 F.3d 1179 (Fed. Cir. 2018) .....*passim*

Oracle Am., Inc. v. Google Inc.,
2016 WL 3181206 (N.D. Cal. June 8, 2016)..... 19

Sega Enters. Ltd. v. Accolade, Inc.,
977 F.2d 1510 (9th Cir. 1992).....3, 5, 13

Seltzer v. Green Day, Inc.,
725 F.3d 1170 (9th Cir. 2013)..... 19

Sony Computer Entm't, Inc. v. Connectix Corp.,
203 F.3d 596 (9th Cir. 2000)..... 21

Sony Corp. v. Universal City Studios, Inc.,
464 U.S. 417 (1984)..... 12

Swatch Grp. Mgmt. Servs. Ltd. v. Bloomberg L.P.,
756 F.3d 73 (2d Cir. 2014) 20

Twentieth Century Music Corp. v. Aiken,
422 U.S. 151 (1975)..... 5

Universal City Studios, Inc. v. Corley,
273 F.3d 429 (2d Cir. 2001) 21

STATUTES

17 U.S.C. § 102(b) 13

OTHER AUTHORITIES

Clark D. Asay, *Software's Copyright Anticommons*,
66 Emory L.J. 265 (2017).....8, 21, 22

Brief for the United States as Amicus Curiae, *Google Inc. v. Oracle America, Inc.*, 135 S. Ct. 2887 (2015) (No. 14-410)..... 18

BSA | The Software Alliance, *The $1 Trillion Economic Impact of Software* (2016), *available at* https://softwareimpact.bsa.org/pdf/Economic_Impact_of_Software_Report.pdf 2

Donald S. Chisum et al., *Last Frontier Conference Report on Copyright Protection of Computer Software*, 30 Jurimetrics 15 (1989) 3

Joseph Farrell et al., *Standard Setting, Patents, and Hold-Up*, 74 Antitrust L.J. 603 (2007)..... 16

Fed. Trade Comm'n, *The Evolving IP Marketplace: Aligning Patent Notice and Remedies with Competition* (2011), *available at* https://www.ftc.gov/sites/default/files/documents/reports/evolving-ip-marketplace-aligning-patent-notice-and-remedies-competition-report-federal-trade/110307patentreport.pdf 7

Fed. Trade Comm'n, *To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy* (2003), *available at* https://www.ftc.gov/sites/default/files/documents/reports/promote-innovation-proper-balance-competition-and-patent-law-and-policy/innovationrpt.pdf..... 8

Final Report of the Nat'l Commission on New Technological Uses of Copyrighted Works (1978) ... 2

Joseph Gratz & Mark A. Lemley, *Platforms and Interoperability in* Oracle v. Google, 31 Harv. J. Law & Tech. 603 (2018) 15, 16

Michael L. Katz & Carl Shapiro, *Antitrust in Software Markets*, *in* Competition, Innovation and the Microsoft Monopoly (1999) 5

Pierre N. Leval, *Toward a Fair Use Standard*, 103 Harv. L. Rev. 1105 (1990) 6, 21

Fred von Lohmann, *The New Wave: Copyright and Software Interfaces in the Wake of* Oracle v. Google, 31 Harv. J. L. & Tech. 517 (2018) 11

Peter S. Menell, *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, 31 Harv. J. L. & Tech. 305 (2018)....10, 11

William F. Patry, *Patry on Fair Use* (May 2018 Update) 21

Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308 (1994)..... 22

Pamela Samuelson, *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, 31 Berkeley Tech. L.J. 1215 (2016)..... 17

Pamela Samuelson, *The Uneasy Case for Software Copyrights Revisited*, 79 Geo. Wash. L. Rev. 1746 (2011) 3

Carl Shapiro & Hal R. Varian, *Information Rules* (1999) 9, 10, 11

Scott A. Sher & Bradley T. Tennis, *Exploiting Others' Investments in Open Standards*, Comp. Pol'y Int'l Antitrust Chron. 1, No. 1 (2016), https://www.competitionpolicyinternational.com/exploiting-others-investments-in-open-standards/ 8

StatCounter, Global Stats, *Mobile System Market Share Worldwide* (Dec. 2019), https://gs.statcounter.com/os-market-share/mobile/worldwide 2

Testimony of Carl Shapiro, Antitrust Modernization Commission, *Exclusionary Conduct* (Sept. 29, 2005) 9

U.S. Copyright Office, *Software-Enabled Consumer Products* (Dec. 2016), *available at* https://www.copyright.gov/policy/software/software-full-report.pdf ..... *passim*

INTEREST OF AMICUS CURIAE

The American Antitrust Institute ("AAI") is an independent non-profit organization devoted to promoting competition that protects consumers, businesses, and society.[1] See http://www.antitrustinstitute.org. AAI serves the public through research, education, and advocacy on the benefits of competition and the use of antitrust enforcement as a vital component of national and international competition policy. AAI also seeks to ensure that intellectual property laws are interpreted and applied in a manner that reflects their ultimate goals of promoting innovation, competition, and consumer welfare.

AAI submits this brief because the Federal Circuit's application of the copyright laws to computer software interfaces undermines those goals and threatens substantial competitive harm in software-dependent markets throughout the U.S. economy. The public is harmed when even a large company like Google must pay royalties to license software interfaces. But this case also has implications for whether start-up firms that may challenge entrenched incumbents (like Google itself) will be deterred from doing so because of the barrier to entry created by the Federal Circuit's overprotection of software interfaces.

---

[1] All parties consent to the filing of this brief. No counsel for a party has authored this brief in whole or in part, and no person other than amicus curiae has made a monetary contribution to fund its preparation or submission. Individual views of members of AAI's Board of Directors or Advisory Board may differ from AAI's positions.

INTRODUCTION

This case will determine the copyrightability and fair use of declaring code that applications developers use to call forth methods of implementing commonly performed computing tasks on the Android mobile platform. The decision will significantly affect the allocation of profits and rents collected on the mobile operating system running on an estimated 74% of the world's smartphones. StatCounter, Global Stats, *Mobile System Market Share Worldwide* (Dec. 2019). And to the extent the Court determines the copyrightability and fair use of software interfaces generally, it will indirectly set national competition policy in software-dependent markets, which are pervasive in the U.S. economy. *See, e.g.*, BSA | The Software Alliance, *The $1 Trillion Economic Impact of Software* 3–4 (2016); U.S. Copyright Office, *Software-Enabled Consumer Products* 3 (December 2016) (noting that "[s]oftware is now nearly ubiquitous," including in consumer products).

Congress extended copyright protection to software in 1980 as a compromise among possible alternatives. Liberalizing patent protection, the availability of which was then unclear, would have gone too far. Defining a new, *sui generis* protection threatened to upset traditions of overarching patent and copyright laws. And affording no protection would have required the software industry to rely on contract, trade secret, or other state laws. *See Final Report of the Nat'l Commission on New Technological Uses of Copyrighted Works* 16–19 (1978).

In the first decade after Congress made its choice, a group of leading intellectual property scholars observed that "Congress . . . has left to the courts the

difficult task[] of determining how to apply copyright to computer programs," and "[c]ourts have generally articulated traditional copyright standards for determining the scope of protection." Donald S. Chisum et al., *Last Frontier Conference Report on Copyright Protection of Computer Software*, 30 Jurimetrics 15, 16–17 (1989). But applying concepts designed for traditional literary works to computer software can be like trying "to fit the proverbial square peg in a round hole." *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992) (internal quotation omitted); *see Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 820 (1st Cir. 1995) (Boudin, J., concurring) (likening difficulties of applying copyright law to computer programs to "assembling a jigsaw puzzle whose pieces do not quite fit").

The problem is that "computer programs are, in essence, utilitarian articles—articles that accomplish tasks." *Sega*, 977 F.2d at 1524; *Lotus*, 49 F.3d at 819 (Boudin, J., concurring) ("The computer program is a *means* for causing something to happen; it has a mechanical utility, an instrumental role, in accomplishing the world's work."). Thus, "[c]omputer programs pose unique problems for the application of the 'idea/expression distinction' that determines the extent of copyright protection." *Sega*, 977 F.2d at 1524. More generally, as Judge Boudin explained, "[u]tility does not bar copyright (dictionaries may be copyrighted), but it alters the calculus" for intellectual property protection. *Lotus*, 49 F.3d at 819. The benefit may be similar (stimulating the production of computer software),[2] "[b]ut the 'cost' side of the equation

---

[2] *But see* Pamela Samuelson, *The Uneasy Case for Software Copyrights Revisited*, 79 Geo. Wash. L. Rev. 1746, 1776 (2011)

may be different [than for traditional literary works] where one places a very high value on public access to a useful innovation that may be the most efficient means of performing a given task." *Id.*

In particular, the calculus for protecting computer software "interfaces" like the Java SE declarations at issue here or the command menu hierarchy at issue in *Lotus* is problematic at best. As Judge Boudin explained:

> Requests for the protection of computer menus present the concern with fencing off access to the commons in an acute form. A new menu may be a creative work, but over time its importance may come to reside more in the investment that has been made by *users* in learning the menu and in building their own mini-programs—macros—in reliance upon the menu. Better typewriter keyboard layouts may exist, but the familiar QWERTY keyboard dominates the market because that is what everyone has learned to use.

Id. at 819–20.

SUMMARY OF ARGUMENT

The Federal Circuit held that software interfaces like the Java SE declarations at issue are entitled to copyright protection and that fair use cannot apply to software innovation built on a copyrighted interface unless the innovation changes the meaning or expression of the copied elements. *Oracle Am., Inc. v. Google*

(identifying "significant developments in the software industry [that] raise questions about how important copyright protection now is to enabling developers to recoup their R&D investments in software").

Inc. (*Oracle I*), 750 F.3d 1339 (Fed. Cir. 2014); *Oracle Am., Inc. v. Google LLC* (*Oracle II*), 886 F.3d 1179 (Fed. Cir. 2018). Those rulings fail to consider that copyright law seeks to promote innovation and consumer welfare by preserving a balance between exclusive rights and competition. *E.g., Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156 (1975).

By ignoring the competition side of the equation—indeed, by rejecting the relevance of interoperability considerations—the Federal Circuit's rulings, if not overturned, may slow innovation and competition in software-dependent markets. *See* Michael L. Katz & Carl Shapiro, *Antitrust in Software Markets*, *in* Competition, Innovation and the Microsoft Monopoly 33 (1999) ("In the absence of compatibility, markets may tip. Hence, the very nature of competition is fundamentally affected[,] . . . setting up a natural tension over the control of interfaces"). The rulings also fail to reflect that copyright precedents applicable to traditional literary and similar works cannot be applied woodenly to computer software. *See Sega*, 977 F.2d at 1524.

1. Determining the copyrightability and fair use of software interfaces should be understood as an exercise in calibrating the contestability of software-driven markets controlled by dominant incumbent firms. Software pervades U.S. commerce, and when software-driven markets tip toward a single firm, affording copyright protection to software interfaces necessary for interoperability threatens to entrench such firms because of network effects, switching costs, and lock-in.

2. The Federal Circuit erroneously relegated several of copyright law's core competition safeguards in software markets—Section 102(b) of the Copyright

Act, the merger doctrine, and fair use—to insignificance. By holding that interoperability concerns are irrelevant to Section 102(b), that merger analysis applies only to design choices made by system creators at the moment of creation, and that Google's compatibility goals were either neutral or weighed *against* a finding of fair use, the court failed to recognize that the idea/expression dichotomy embodied in copyright's limiting doctrines is designed to promote competition and enable innovators to create interoperable products and services.

3. The Federal Circuit's holding that, no matter how innovative new software that copies declaring code, it does not qualify as a transformative use if "there are no changes to the expressive content or message" of the copied elements, *Oracle II*, 886 F.3d at 1201–02, guts the fair-use doctrine as applied to software. This Court should clarify that a "transformative use is one that communicates something new and different from the original *or expands its utility*." *Authors Guild v. Google, Inc.*, 804 F.3d 202, 214 (2d Cir. 2015) (Leval, J.) (emphasis added); *see* Pierre N. Leval, *Toward a Fair Use Standard*, 103 Harv. L. Rev. 1105 (1990) (fair use requires "disciplined focus on the utilitarian, public-enriching objectives of copyright"); *see id*. at 1108 & nn.14 & 17 (utilitarian purpose confirmed by *The Federalist* No. 43 and the Statute of Anne of 1709). Failing to recognize utilitarian transformations in the software context is perverse, because software's benefit is primarily functional. And software interfaces become standards (and are copied) because of their functional, not expressive, value. Moreover, pairing interfaces like the Java SE declarations with new implementing code changes the "message" of the declarations.

ARGUMENT

I. AFFORDING COPYRIGHT PROTECTION TO SOFTWARE INTERFACES CAN ENTRENCH DOMINANT FIRMS PROTECTED BY NETWORK EFFECTS

A. Copyrights on Software Interfaces Risk Lock-In and Holdup

The risk of harmful competitive effects of *patents* on software or other technologies that become elements of industry standards is well known. *See generally* Fed. Trade Comm'n, *The Evolving IP Marketplace: Aligning Patent Notice and Remedies with Competition* 234 (2011). Product manufacturers can become locked in to the standard and thereby susceptible to patent "holdup," with the result that royalties are excessive and innovation by manufacturers is discouraged. *See id.* at 227. Courts have adjusted patent remedies to avoid such holdup. *See Commonwealth Sci. & Indus. Research Org. v. Cisco Sys., Inc.*, 809 F.3d 1295, 1305 (Fed. Cir. 2015); *Apple, Inc. v. Motorola, Inc.,* 757 F.3d 1286, 1331–32 (Fed. Cir. 2014), *overruled on other grounds by Williamson v. Citrix Online, LLC*, 792 F.3d 1339 (Fed. Cir. 2015) (en banc); *see also eBay Inc. v. MercExchange, L.L.C.*, 547 U.S. 388, 396 (2006) (Kennedy, J., concurring) (cautioning against injunctive relief "[w]hen the patented invention is but a small component of the product the companies seek to produce and the threat of an injunction is employed simply for undue leverage in negotiations").

A similar problem arises with copyrighted software interfaces. Copyright on largely functional elements of software that become an industry standard gives a copyright holder anticompetitive power to

thwart or tax innovative developments that build upon the elements, and to misappropriate for itself investments by users or developers in learning those elements. *Cf. Lotus*, 49 F.3d at 821 ("[I]t is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an investment in learning made by the users and not by Lotus."). Even if the copyrighted elements are not as essential and the lock-in not as severe as with a standard-essential patent, the anticompetitive harm from a copyright holder's ability to raise the costs of the innovative developments—to the detriment of new entrants, customers of the incumbent, and the public at large—is similar.

The risk of copyright holdup seems likely to increase as software development becomes increasingly collaborative and "any given piece of software may include dozens, hundreds, or even thousands of copyright holders." Clark D. Asay, *Software's Copyright Anticommons*, 66 Emory L.J. 265, 279 (2017). The "building-block approach to software development . . . means that some copyright holder of a software object within a particular software stack could become an obstacle to the entire stack's use." *Id.* at 314; *cf.* Fed. Trade Comm'n, *To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy*, ch. 3, at 44 (2003) ("innovation [in software] occurs cumulatively"). And that risk is exacerbated by owners' readiness to employ "'[o]pen early, closed late' strategies based on copyright." Scott A. Sher & Bradley T. Tennis, *Exploiting Others' Investments in Open Standards*, Comp. Pol'y Int'l Antitrust Chron. 1, No. 1, at 3–4 (2016) ("[T]herein lies the danger: that a firm will employ an open policy in order to gain dominance and then impose less favorable interconnection terms once

dominance has been achieved.") (quoting Testimony of Carl Shapiro, Antitrust Modernization Commission, *Exclusionary Conduct* 15-16 (Sept. 29, 2005)).

B. Lock-In Can Cement Software-Based Monopolies

The lock-in caused by the overprotection of software interfaces is particularly anticompetitive because it tends to prevent new entrants from challenging dominant incumbent firms in platform-driven industries.

The multisided desktop and mobile operating system markets that serve as the backdrop to this case are illustrative. According to "Metcalfe's Law," the proportional value to a network of a user's investment in joining the network is the square of the number of users who do so, such that "a tenfold increase in the size of the network leads to a hundredfold increase in its value." Carl Shapiro & Hal R. Varian, *Information Rules* 184 (1999). In Shapiro & Varian's example, if a network that has a $1 value to a single user increases to 10 users, then the *network*'s total value increases to *$100*. *Id*. In an operating system environment, both consumers and software developers (as well as hardware and other complementors) invest in learning system software, adding several different dimensions of value to the network.

Software-based markets are characterized by strong positive network effects, which means lock-in increases over time because switching costs increase as the network size increases and network participants make greater investments in training to learn the system. *See id*. at 121 ("[T]he training costs associated with replicating one's proficiency with a familiar piece of software tend to grow the more experience

one has with the familiar program.").[3] New entrants seeking to introduce a rival operating system must overcome the costs of inducing *both* consumers and software developers (as well as complementors) to switch to the new network. *Id*. at 184 ("The challenge to companies seeking to introduce new but incompatible technology into the market is to build network size by overcoming the *collective switching costs*—that is, the combined switching costs of all users."); Peter S. Menell, *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, 31 Harv. J. L. & Tech. 305, 458 (2018) ("companies seeking to leapfrog a widely adopted standard face substantial risk" and must not only invent a better platform and devise a strategy to migrate consumers away from the dominant platform, but also "encourag[e] other software and complementary product developers to build for the new platform").

Accordingly, "[i]n many information industries, collective switching costs are the biggest single force working in favor of incumbents." Shapiro & Varian, *supra*, at 184; *see id*. at 185–86 (explaining that inefficient QWERTY keyboard layout persists because "the *human* component of the system" raises collective switching costs and creates significant difficulties for coordinating a move to superior technology). And "[w]orse yet for would-be entrants and innovators, switching costs work in a nonlinear way: convincing ten people connected in a network to switch to your

[3] Training costs are not the only switching costs created by network effects. Investments in the software itself (apart from training), hardware, durable complementary assets, and information systems also give rise to switching costs. Shapiro & Varian, *supra*, at 184.

incompatible network is more than ten times as hard as getting one customer to switch. But you need all ten, or most of them: no one will want to be the first to give up the [incumbent] network externalities and risk being stranded." *Id.* at 184–85.

In an important sense, then, the Court's decision whether to let stand the Federal Circuit's rulings will affect whether mature, software-driven markets with strong incumbents are contestable by entrepreneurs. "Leaving API design specifications outside of copyright protection enables entrepreneurs seeking to improve on successful platforms to build bridges for users and programmers," which "avoids excess inertia and accommodates creative destruction and evolution in those areas where the proprietor of the standard platform lacks patent protection." Menell, *supra*, at 468. However, if copyright owners can appropriate developers' training investments by asserting copyright protection over interfaces, then collective switching costs can make it virtually impossible for entrepreneurial rival networks to launch, grow, and eventually challenge established incumbents. *See* Shapiro & Varian, *supra*, at 184, 195 ("[S]ometimes this kind of barrier can be insurmountable. Incumbents with intellectual property rights over an older generation of technology may have the ability to unilaterally blockade a migration path . . . [and] stop rivals in their tracks[.]").

The paradigm is not limited to desktop and mobile operating systems, but rather applies wherever a dominant incumbent asserts copyright protection over the functional aspects of software interfaces. *See* Fred von Lohmann, *The New Wave: Copyright and Software Interfaces in the Wake of* Oracle v. Google, 31 Harv. J. L. & Tech. 517, 519-27 (2018) (discussing spate of

copyright assertions over interfaces in the wake of this case in markets for enterprise statistical software, electronic design automation software, network switch operating system software, and digital cinema software).

II. COPYRIGHT'S LIMITING DOCTRINES ARE DESIGNED TO PRESERVE INNOVATORS' ABILITY TO CREATE INTEROPERABLE AND COMPETITIVE PRODUCTS AND SERVICES

The Constitution's intellectual property clause contemplates both "the need to encourage innovation and the avoidance of monopolies which stifle competition without any concomitant advance in the 'Progress of Science and useful Arts.'" *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 146 (1989). Thus "the Clause contains both a grant of power and certain limitations upon the exercise of that power." *Id*. It assigns Congress the task of striking "a difficult balance between the interests of authors and inventors in the control and exploitation of their writings and discoveries on the one hand, and society's competing interest in the free flow of ideas, information, and commerce on the other hand." *Sony Corp. v. Universal City Studios, Inc.*, 464 U.S. 417, 429 (1984).

To achieve this balance in practice, Congress and the courts have introduced a variety of competition safeguards that limit the scope of copyright protection, including § 102(b) of the Copyright Act, merger, and fair use. But instead of ensuring that the anticompetitive harm threatened by copyright protection of software interfaces is cabined by a sufficiently liberal reading of these limiting doctrines, the Federal Circuit relegated each of them to insignificance. The court

held that interoperability is irrelevant to copyrightability under Section 102(b), *Oracle I*, 750 F.3d at 1368, that merger analysis applies only to design choices made by system creators at the moment of creation, *id.* at 1361, and that Google's compatibility goals were either neutral or weighed *against* a finding of fair use, *Oracle II*, 886 F.3d at 1207. This was error.

A. Section 102(b)

Section 102(b) of the Copyright Act codifies the "idea/expression dichotomy"—the basic principle that copyright law protects creative expression but not the underlying concepts embodied in copyrighted works.[4] Without this distinction, a copyright owner could wrongly obtain "a *de facto* monopoly over the functional aspects of his work—aspects that were expressly denied copyright protection by Congress." *Sega*, 977 F.2d at 1526. To gain a lawful monopoly over such functional aspects, "the creator of the work must satisfy the more stringent standards imposed by the patent laws." *Id.* Copyright protection, unlike patent protection, "leav[es] the ideas, facts, and functional concepts in the public domain for others to build on." *Id.* at 1527.

In rejecting the invocation of interoperability concerns in copyrightability analysis, *Oracle I,* 750 F.3d at 1368–72, the Federal Circuit failed to adequately grasp that Section 102(b) employs the idea/expression dichotomy to preserve competition and innovation by ensuring public access to the functional concepts

---

[4] 17 U.S.C. § 102(b) ("In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form on which it is described, explained, illustrated, or embodied in such work.").

embodied in copyrighted works. The court emphasized that "the declaring code could have been written and organized in any number of ways and still have achieved the same functions." *Id*. at 1368; *see also* Br. for the United States as Amicus Curiae at 13 ("U.S. Br."). But this is no answer to the question of whether the declaring code is unprotectable under Section 102(b) as a functional system or method of operation. *See, e.g., Bikram's Yoga College of India v. Evolation Yoga, LLC*, 803 F.3d 1032, 1042 (9th Cir. 2015) ("It makes no difference that similar results could be achieved through a different organization . . . . [T]he possibility of attaining a particular end through multiple different methods does not render the uncopyrightable a proper subject of protection."); *Lotus*, 49 F.3d at 818 ("That programs can offer users the ability to write macros in many different ways does not change the fact that . . . the Lotus menu command hierarchy is a 'method of operation.'").

As other courts have recognized, the fact that copying a software interface is necessary to achieve interoperability or compatibility, and that a defendant has copied no more than is necessary to do so, tends to prove that a plaintiff's infringement claim falls on the wrong side of the idea/expression dichotomy for purposes of clearing the bar set by Section 102(b). *See, e.g., Lotus*, 49 F.3d at 817 ("That the Lotus command menu hierarchy is a 'method of operation' becomes clearer when one considers program compatibility."); *cf. Computer Associates International, Inc. v. Altai*, 982 F.2d 693, 708 (2d Cir.1992) ("In the context of computer program design, the concept of efficiency is akin to deriving the most concise logical proof or formulating the most succinct mathematical computation. Thus, the more efficient a set of modules are, the more

closely they approximate the idea or process embodied in that particular aspect of the program's structure.").

The U.S Copyright Office recognized as much in a recent report on copyright issues related to software-enabled consumer products. *Software-Enabled Consumer Products*, *supra*. In the report, the Office "recognizes the importance of preserving the ability to develop products and services that can interoperate with software-enabled consumer products, and the goal of preserving competition in the marketplace." *Id*. at 52. Yet, it concluded that legislation was not needed to achieve these goals because "faithful application of existing copyright law doctrines can preserve the twin principles of interoperability and competition." *Id*.

Among other things, the Office observed that Section 102(b) exempts methods of operation *embodied or described* in computer code from copyright protection," and consequently "the Act does not prevent a competitor from studying code to determine the underlying methods it teaches, and from implementing those methods using *different* code than the original, to create an interoperable or competitive software-enabled consumer product." *Id*. at 53 (emphases in original).

By way of illustration, the Office described one "famous example" whereby Phoenix Technologies used a clean-room process to engineer a compatible basic input-output system (BIOS) for IBM personal computers. *Id*. at 53. Phoenix Technologies reimplemented the IBM BIOS API in part by "copying only the elements necessary for compatibility." Joseph Gratz & Mark A. Lemley, *Platforms and Interoperability in* Oracle v. Google, 31 Harv. J. Law & Tech. 603, 611 (2018). And "As with Java, those [copied] elements included a hierarchy of commands—for example, all

calls beginning with '0x10' related to video services, and within that category the call '0x10 0x9H' would write a particular letter to the screen." *Id*.

B. Merger

The Copyright Office also observed that the doctrine of merger is "a promising avenue to permit copying for purposes of interoperability, at least in the narrow circumstances in which [it] appl[ies]." *Software-Enabled Consumer Products*, *supra*, at 54. When an idea can be expressed only in a limited number of ways, the merger doctrine renders the expression unprotectable to ensure the author does not receive "a monopoly over the idea itself, thereby preventing others from using that same idea in other works." *Id*. at 15.

However, the Federal Circuit's holding eliminates this promise in the software context, leaving no room for application of the merger doctrine to counteract copyright holdup. The Federal Circuit formalistically applied *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521 (9th Cir. 1984), to assess the copyrightability of the declarations as of the time that Sun originally created them, before any developer had ever written in Java SE. *Oracle I*, 750 F.3d at 1361; *see also* U.S. Br. at 13-14. But neither the Court nor the government grappled with *Altai*'s observation that efficiency in computer program design is akin to deriving the most concise logical proof or most succinct mathematical computation. 982 F.2d at 708.

No product is a standard, whether *de jure* or *de facto*, at inception. When copyrighted software interfaces become standards, the efficiency derived from the interface is necessarily an emergent property. *See* Joseph Farrell et al., *Standard Setting, Patents, and*

Hold-Up, 74 Antitrust L.J. 603, 608, n.26 (2007) ("A standard could be built around initially arbitrary choices that become essential once the standard is established," and "[e]x ante arbitrary choices (with minimal ex ante advantage) may be most likely where a copyright, rather than a patent, is involved"). The Federal Circuit ignored that systems give rise to emergent mathematical facts and methods, and instead evaluated Java's elements exclusively at the time that Java was created. By doing so, it artificially read hold-up concerns out of Section 102(b) entirely, improperly placing non-statutory limits on the idea/expression dichotomy. *Cf.* Pamela Samuelson, *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, 31 Berkeley Tech. L.J. 1215, 1278 (2016) ("Anyone who develops an API for achieving program interoperability is, in effect, . . . creating constraints on the design choices of all others who want to [subsequently] develop programs to interoperate with his platform . . . ."); *id.* at 1282 (noting that CONTU accepted the proposition "that function and expression might merge over time"); *Lotus,* 49 F.3d at 819 (new computer menu may be creative at inception but gain importance over time because of user investments).

C. Fair Use

The Copyright Office further "believes that, in many cases, copying of appropriately limited amounts of code from one software-enabled product into a competitive one for purposes of compatibility and interoperability should also be found to be a fair use." *Software-Enabled Consumer Products, supra*, at 57; *see id.* at 59 ("proper application of [fair use] principles should ensure that copyright law preserves the ability to create interoperable products and services").

To be sure, the Federal Circuit said that concerns about interoperability may be relevant to fair use. *See Oracle I*, 750 F.3d at 1372, 1377. So did the United States. *See* Brief for the United States as Amicus Curiae 17, *Google Inc. v. Oracle America, Inc.*, 135 S. Ct. 2887 (2015) (No. 14-410) (interoperability and lock-in concerns are "substantial and important" but "are far better addressed through the fair-use doctrine"); *see also Lotus*, 49 F.3d at 821 (Boudin, J., concurring) (suggesting that fair use was alternative, albeit inferior, doctrinal hook to ensure that users are not locked into *de facto* standards).

Yet, in its fair-use decision, the Federal Circuit dismissed interoperability considerations. The court framed Google's compatibility argument as, "Google sought 'to capitalize on the fact that software developers were already trained and experienced in using the Java API packages at issue.'" *Oracle II,* 886 F.3d at 1206 (quoting *Oracle I*, 750 F.3d at 1371). "But," the court said, "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." *Id.* at 1206–07; *see also* U.S. Br. at 20.

The Federal Circuit misapprehended the compatibility point. It is not about free-riding, but whether the public is served insofar as copying the Java SE declarations gives developers "an option to exploit their own prior investment in learning" the declarations rather than remain captives of the copyright owner. *Lotus*, 49 F.3d at 821 (Boudin, J., concurring); *see Lotus*, 49 F.3d at 817–18 (finding "absurd" Lotus's theory that "if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used"); *id.* at 818 (rejecting rule whereby "the user would have to rewrite

his or her macro [to perform a certain operation in another program] using that other program's menu command hierarchy. . . despite the fact that the macro is clearly the user's own work product."). By the Federal Circuit's own prior reckoning in *Oracle I*, fostering interoperability of use should have been at least *relevant* to fair use.

III. THE COURT SHOULD SANCTION UTILITARIAN TRANSFORMATIONS OF SOFTWARE INTERFACES AS FAIR USE

The Federal Circuit also failed to heed the Court's admonition that "the goal of copyright, to promote science and the arts, is generally furthered by the creation of transformative works." *Campbell v. Acuff-Rose Music, Inc.,* 510 U.S. 569, 579 (1994). The court held that, no matter how innovative the new software, it does not qualify as a transformative use if "there are no changes to the expressive content or message" of the elements that are copied. *Oracle II*, 886 F.3d at 1201–02. As the district court noted, "[i]f this were enough to defeat fair use, it would be impossible ever to duplicate declaring code as fair use." *Oracle Am., Inc. v. Google Inc.*, 2016 WL 3181206, at *8 (N.D. Cal. June 8, 2016).

The Federal Circuit relied on *Seltzer v. Green Day, Inc.*, 725 F.3d 1170 (9th Cir. 2013), for the proposition that "a work is not transformative where the user 'makes no alteration to the *expressive content or message* of the original work.'" *Oracle II*, 886 F.3d at 1201 (quoting *Seltzer*, 725 F.3d at 1177). But *Seltzer* actually said, "In the *typical* 'non-transformative' case, the use is one which makes no alteration to the expressive content or message of the original work." *Seltzer*, 725 F.3d at 1177 (emphasis added and omitted). *Seltzer*

did not involve software code, nor did any of the other cases the Federal Circuit cited.

The Court should clarify that works also can be transformative if they expand the utility of copyrighted works. *See, e.g., Authors Guild v. Google, Inc.*, 804 F.3d 202, 214 (2d Cir. 2015) (Leval, J.) ("transformative use is one that communicates something new and different from the original *or expands its utility*") (emphasis added); *Fox News Network, LLC v. TVEyes, Inc.*, 883 F.3d 169, 176 (2d Cir.), *cert. denied*, 139 S. Ct. 595 (2018) (quoting same); *see also Swatch Grp. Mgmt. Servs. Ltd. v. Bloomberg L.P.*, 756 F.3d 73, 84 (2d Cir. 2014) ("a secondary work 'can be transformative in function or purpose without altering or actually adding to the original work.'") (quoting *A.V. ex rel. Vanderhye v. iParadigms, LLC*, 562 F.3d 630, 639 (4th Cir. 2009)).

The Federal Circuit's failure to recognize this point in the context of computer software is perverse.[5] While expressive components of software may be protected by copyright (subject to § 102(b)), software's benefit is primarily functional and utilitarian, not artistic. And software interfaces become standards (and are copied) because of their functional, not expressive, value. Not recognizing utilitarian transformations would enable the holder of a software interface copyright with the barest degree of expressive creativity to

[5] The Federal Circuit did acknowledge that placing a copyrighted work in a new context to serve a different purpose may be transformative. *Oracle II*, 886 F.3d at 1202. But it concluded that copying elements of a software program to develop a new operating system for a new category of products (smartphones) would not serve a different purpose. If not *ipse dixit*, this conclusion can only be explained by the court's giving dispositive weight to whether there is a change in message.

monopolize (or tax) broad swaths of commerce that incorporate the interface and would thwart the most significant, pro-competitive uses of the fair-use doctrine in software-dependent industries. *Cf.* Leval, *supra*, at 1134 ("the court should focus on harm to the plaintiff's interest *as copyright owner*" and inquire whether he has suffered "the types of harms against which the copyright law protects"); William F. Patry, *Patry on Fair Use* § 6:7 (May 2018 Update) (copyright only concerned with harm "caused by the use of expression"); U.S. Copyright Office, *Software-Enabled Consumer Products, supra*, at 57 (question is "whether the use is principally for the purpose of exploiting the creativity of the original author of the code"; "interoperability is a favored purpose") (internal quotation marks omitted); *id*. at 58 ("literal copying of code may be favored, if the purpose is simply to 'permit . . . functionality' of a software-enabled device, and not to exploit the creativity of the original author").

Moreover, although the Federal Circuit claimed otherwise, using declaring code to call forth different implementing code changes the "message" of the declaring code. *Cf. Universal City Studios, Inc. v. Corley*, 273 F.3d 429, 446–49 (2d Cir. 2001) (recognizing computer code itself as a form of speech); *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596, 606–07 (9th Cir. 2000) ("Connectix's drafting of entirely new object code for its VGS program [is] transformative, despite the similarities in function and screen output."). As Professor Asay points out, "Software interfaces" like the Java SE declarations "are strictly functional in carrying out the specified functions and facilitating communication between software products. . . . Hence, whatever creativity interfaces entail only becomes present and relevant when they are paired with the

software that implements them." Asay, *supra*, at 321; *cf.* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308, 2317 (1994) ("No one would want to buy a program that did not behave, i.e., that did nothing, no matter how elegant the source code 'prose' expressing that nothing.").

The Federal Circuit's fair-use ruling prevents the fair-use doctrine from acting as a safety valve "to avoid rigid application of the copyright statute [in the software context] when . . . it would stifle the very creativity which that law is designed to foster." *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 550 n.3 (1985) (internal quotation omitted). Together with its copyrightability holding, the court has tilted the balance contemplated by the Constitution's intellectual property clause and the idea/expression dichotomy to the detriment of consumers, developers, entrepreneurs, and challengers to dominant incumbents in a wide swath of U.S. markets.

This Court should recognize an appropriate role for copyright's limiting doctrines to promote competition and consumer welfare in software markets by allowing innovators to use software interfaces to create interoperable products and services.

CONCLUSION

For the foregoing reasons, the copyrightability and fair use decisions of the Federal Circuit should be reversed.

Respectfully submitted,

RANDY M. STUTZ
Counsel of Record
AMERICAN ANTITRUST INSTITUTE
1025 Connecticut Ave., NW
Suite 1000
Washington, DC 20036
(202) 905-5420
rstutz@antitrustinstitute.org