

No. 18-956

IN THE
Supreme Court of the United States

GOOGLE LLC,
Petitioner,

v.

ORACLE AMERICA, INC.,
Respondent.

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

**BRIEF OF MICROSOFT CORPORATION AS
AMICUS CURIAE IN SUPPORT OF PETITIONER**

LISA W. BOHL
MOLOLAMKEN LLP
300 N. LaSalle St.
Chicago, IL 60654
(312) 450-6700

JEFFREY A. LAMKEN
Counsel of Record
MICHAEL G. PATTILLO, JR.
WILLIAM J. COOPER
MOLOLAMKEN LLP
The Watergate, Suite 660
600 New Hampshire Ave., NW
Washington, D.C. 20037
(202) 556-2000
jlamken@mololamken.com

Counsel for Amicus Curiae

TABLE OF CONTENTS

	Page
Interest of <i>Amicus Curiae</i>	1
Reasons for Granting the Petition	3
I. A Flexible Fair Use Doctrine Is Critical To Balancing the Interests of Copyright Protection and Follow-On Innovation in Software.....	5
A. Software Encompasses Collabora- tive, Functional Elements Not Present in Traditional Creative Works Subject to Copyright Protection	5
B. A Flexible Fair Use Doctrine Is Essential To Accommodate the Modern Software Development Approach—As Courts Have Long Recognized	9
C. Experience Has Shown That a Flexible Fair Use Doctrine Fosters Innovation in Computer Software	12
II. The Federal Circuit’s Decision Defies Settled Fair-Use Principles and Misapprehends the Nature of the Computer Industry	14
A. The Federal Circuit’s Disregard of the Functional Nature of Oracle’s Declaring Code and SSOs in the Fair-Use Analysis Defies Precedent and Industry Reality	14
B. The Federal Circuit Fundamen- tally Misunderstood What Consti- tutes a “Transformative Use” of Software.....	17

TABLE OF CONTENTS—Continued

	Page
III. The Federal Circuit’s Rigid, Narrow Approach to Fair Use Threatens the Viability of the Interconnected Software Ecosystem	21
Conclusion.....	24

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>Apple Comput., Inc. v. Microsoft Corp.</i> , 35 F.3d 1435 (9th Cir. 1994).....	8
<i>Atari Games Corp. v. Nintendo of Am. Inc.</i> , 975 F.2d 832 (Fed. Cir. 1992)	12
<i>Campbell v. Acuff-Rose Music, Inc.</i> , 510 U.S. 569 (1994).....	<i>passim</i>
<i>Comput. Assocs. Int’l, Inc. v. Altai, Inc.</i> , 982 F.2d 693 (2d Cir. 1992)	5, 7
<i>Dr. Seuss Enters., L.P. v. Penguin Books USA, Inc.</i> , 109 F.3d 1394 (9th Cir. 1997)	16
<i>Feist Publ’ns, Inc. v. Rural Tel. Serv. Co.</i> , 499 U.S. 340 (1991).....	10, 15
<i>Fox News Network, LLC v. TVEyes, Inc.</i> , 883 F.3d 169 (2d Cir. 2018), cert. denied, 139 S. Ct. 595 (2018)	16
<i>Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc.</i> , 964 F.2d 965 (9th Cir. 1992)	11
<i>Lexmark Int’l, Inc. v. Static Control Components</i> , 387 F.3d 522 (6th Cir. 2004)	8
<i>Lotus Dev. Corp. v. Borland Int’l, Inc.</i> , 49 F.3d 807 (1st Cir. 1995)	5, 20
<i>Mattel Inc. v. Walking Mountain Prods.</i> , 353 F.3d 792 (9th Cir. 2003).....	16
<i>Micro Star v. Formgen Inc.</i> , 154 F.3d 1107 (9th Cir. 1998).....	15
<i>Perfect 10, Inc. v. Amazon.com, Inc.</i> , 508 F.3d 1146 (9th Cir. 2007).....	10, 14

TABLE OF AUTHORITIES—Continued

	Page(s)
<i>Sega Enters. Ltd. v. Accolade, Inc.</i> , 977 F.2d 1510 (9th Cir. 1992).....	<i>passim</i>
<i>Sony Comput. Entm't, Inc. v. Connectix Corp.</i> , 203 F.3d 596 (9th Cir. 2000).....	<i>passim</i>
<i>Sony Corp. of Am. v. Universal City Studios, Inc.</i> , 464 U.S. 417 (1984).....	5, 9, 10, 16
<i>Stewart v. Abend</i> , 495 U.S. 207 (1990)	10
<i>Swirsky v. Carey</i> , 376 F.3d 841 (9th Cir. 2004).....	10
<i>Wall Data Inc. v. L.A. Cty. Sheriff's Dep't</i> , 447 F.3d 769 (9th Cir. 2006)	17
 CONSTITUTIONAL PROVISION AND STATUTES	
U.S. Const. art. I, § 8, cl. 8	5, 9, 14, 21
17 U.S.C. § 107.....	2
17 U.S.C. § 107(2)	15
28 U.S.C. § 1295(a)(1).....	21
 OTHER AUTHORITIES	
Clark D. Asay, <i>Software's Copyright Anticommons</i> , 66 Emory L.J. 265, (2017).....	<i>passim</i>
Clark D. Asay, <i>Transformative Use in Software</i> , 70 Stan. L. Rev. Online 9 (2017).....	18, 19
Carliss Baldwin & Eric von Hippel, <i>Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation</i> , 22 Org. Sci. 1399 (2011)	6, 7, 22

TABLE OF AUTHORITIES—Continued

	Page(s)
Jonathan Corbet & Greg Kroah-Hartman, Linux Found., <i>2017 Linux Kernel Development Report</i> (Oct. 25, 2017), https://www.linuxfoundation.org/publications/2017/10/2017-state-of-linux-kernel-development/	6
Joseph Gratz & Mark A. Lemley, <i>Platforms and Interoperability in Oracle v. Google</i> , 31 Harv. J.L. & Tech. 603 (2018).....	7, 12, 23
Mark Kaelin, <i>Microsoft Cognitive Services: Leading the AI Charge</i> , TechRepublic (May 22, 2017), http://www.techrepublic.com/article/build-2017-ai-will-change-everything-and-microsoft-looks-to-lead-the-way/	22
Jamie Rigg, <i>Sony Can't Stop Making Money from PlayStation</i> , Engadget (July 31, 2018), https://www.engadget.com/2018/07/31/sony-playstation-4-millions	13
Karen Rose et al., Internet Soc'y, <i>The Internet of Things: An Overview</i> (2015), https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf	23

TABLE OF AUTHORITIES—Continued

	Page(s)
Jon Russell, <i>Sony Posts \$2.1B Profit as PlayStation Sales Keep on Growing</i> , TechCrunch (Oct. 30, 2018), https://techcrunch.com/2018/10/30/sony-posts-2-1b-profit-as-playstation-sales-keep-on-growing	13
Pamela Samuelson & Clark D. Asay, <i>Saving Software’s Fair Use Future</i> , 31 Harv. J.L. & Tech. 535 (2018)	16
<i>US Video Game Industry Revenue Reaches \$36 Billion in 2017</i> , Entm’t Software Ass’n (Jan. 18, 2018), http://www.theesa.com/article/us-video-game-industry-revenue-reaches-36-billion-2017/	13

IN THE
Supreme Court of the United States

No. 18-956

GOOGLE LLC,
Petitioner,

v.

ORACLE AMERICA, INC.,
Respondent.

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

**BRIEF OF MICROSOFT CORPORATION AS
AMICUS CURIAE IN SUPPORT OF PETITIONER**

INTEREST OF *AMICUS CURIAE*¹

Microsoft Corporation (“Microsoft”) is a leading innovator in computer software; it has been creating software platforms and APIs for application developers for over forty years. Microsoft’s mission is to enable individuals

¹ Pursuant to Supreme Court Rule 37.6, counsel for *amicus curiae* states that no counsel for a party authored this brief in whole or in part. No counsel or party made a monetary contribution intended to fund the preparation or submission of this brief, and no person other than *amicus* or its counsel made such a contribution. All parties have consented to the filing of this brief.

and businesses throughout the world to realize their full potential by creating technology that transforms the ways people work, play, and communicate. Microsoft develops, manufactures, licenses, sells, and supports a wide range of programs, devices, and services, including Windows, Microsoft Azure, Microsoft Office 365, Surface, Xbox and Xbox Live, and Bing. And it invests billions of dollars on research, development, and promotion of new technologies, products, and services to compete in dynamic technology markets.

This case presents important questions concerning the Copyright Act’s authorization of “the fair use of a copyrighted work” under 17 U.S.C. § 107—in particular, how it applies to functional computer code. In overturning the jury’s verdict here, the decision below takes an unduly narrow view of fair use that grants functional code the same level of copyright protection as creative expression in a novel. That has profoundly negative consequences for the computer industry, which depends on a robust fair use doctrine to ensure that software from different vendors will work well together and that developers can create collaborative innovations in software and hardware.

Microsoft has a unique—and balanced—perspective on the legal, economic, and technological issues this case implicates. Microsoft relies on copyright protection to develop and recover its investment in its products and services. Conversely, Microsoft also uses and licenses copyrighted works, and has a longstanding strategic interest in preserving room for legitimate reverse-engineering, competitive analysis, and innovative follow-on development of existing software. Microsoft, its customers, and its developers also need their products to interoperate with copyrighted products provided by

others. To that end, Microsoft must be able to control deployment of its own products and services, while ensuring that its customers and developers can continue to use systems, platforms, infrastructures, and solutions built from connectable offerings provided by multiple vendors. Further, Microsoft uses, contributes to, and sponsors open-source projects, which also rely on settled copyright law—both its protections and exceptions. For example, Microsoft has for several years been the most active corporate maintainer of projects on GitHub, the leading open-source development platform, which Microsoft acquired in 2018. Microsoft also implements APIs from the open-source community in its Windows and Azure products. Microsoft thus has a profound interest in the copyright issues presented in this case.

REASONS FOR GRANTING THE PETITION

This case presents an issue of critical importance to the computer industry—the proper application of copyright law’s “fair use” doctrine to software. Software presents unique challenges for copyright. Copyright provides strong protections for aspects of software that reflect creative expression. But copyright must allow some reuse of software’s functional aspects to enable the collaborative development and interoperability that are critical to technological progress in the computer industry. For decades, courts have addressed those competing interests through a flexible application of the fair use doctrine. Copyright holders and follow-on developers alike have flourished under that approach.

Microsoft urges the Court to grant review because the Federal Circuit’s fair-use decision threatens to upend that careful balance—and the rich, highly interoperable ecosystem of programmers, software, and hardware that brings enormous value to the American public. The

Federal Circuit disregarded the critical significance of the nature of the copyrighted material, affording functional software elements the same level of protection as creative aspects of a work of fiction. The court also imposed a problematically narrow standard for evaluating “transformative use” of functional software code. While Google used the software interfaces at issue for the same purpose as in Oracle’s Java platform—allowing a program to invoke computer functionalities—it incorporated them into a completely different platform that opened new technological possibilities for programmers and consumers. Such innovation promotes the purposes of copyright law, and fair-use analysis should give it due weight.

The Federal Circuit’s decision threatens disastrous consequences for innovation. Software production today is often a highly collaborative process in which many different players participate. The industry’s current practices developed in reliance on decades of court decisions permitting robust fair use of functional software elements. Those cases accommodate the practical need for third parties to access and reuse functional code—like the software interfaces at issue here—to ensure the availability of programmers and to facilitate interoperability across myriad software platforms and hardware devices. Innovations in hardware (such as the “Internet of Things”) and software (such as cloud computing) have made that once-stable body of law increasingly critical.

After hearing the evidence, the jury understood those considerations and concluded that Google’s fair-use defense was valid. The Federal Circuit’s reversal of that verdict as a matter of law threatens fair use’s vitality and extinguishes the necessary “breathing room” for the ecosystem of innovation it protects. This Court should

grant review to ensure that copyright does not impede, rather than “promote,” “the Progress of Science and useful Arts” in the software context. U.S. Const. art. I, §8, cl. 8.

I. A FLEXIBLE FAIR USE DOCTRINE IS CRITICAL TO BALANCING THE INTERESTS OF COPYRIGHT PROTECTION AND FOLLOW-ON INNOVATION IN SOFTWARE

“From its beginning, the law of copyright has developed in response to significant changes in technology.” *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 430 (1984). Software reflected a leap forward in technology. But it also presented new issues for copyright law not posed by traditional literary works. Unlike a novel, software is a “hybrid” of both extremely creative and highly functional elements. *Comput. Assocs. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992). For decades, courts have adapted copyright law to address that reality: They have afforded strong protections to creative aspects of software, while allowing broad reuse of functional software code under a robust fair use doctrine. The computer industry has flourished under that approach, to the benefit of copyright holders and third parties alike. The Federal Circuit’s decision upends that approach and threatens the technological innovation it fostered.

A. Software Encompasses Collaborative, Functional Elements Not Present in Traditional Creative Works Subject to Copyright Protection

Although “[m]ost of the law of copyright * * * developed in the context of literary works such as novels, plays, and films,” “[t]he problem presented by computer programs is fundamentally different.” *Lotus Dev. Corp.*

v. *Borland Int'l, Inc.*, 49 F.3d 807, 819 (1st Cir. 1995) (Boudin, J., concurring). Software is made differently from, and serves different purposes than, traditional literary works.

1. Unlike a novel, software today typically is not produced by a single author. Instead, “more and more software is collaboratively built.” Clark D. Asay, *Software’s Copyright Anticommons*, 66 Emory L.J. 265, 279 (2017). Under the previously dominant “producer” model, a single company creates a “proprietary software offering.” *Id.* at 284. That model still serves an important role in the computer industry. But now, “open collaborative innovation projects,” including open-source models, also serve as important “sources of innovative products, processes, and services.” Carliss Baldwin & Eric von Hippel, *Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation*, 22 *Org. Sci.* 1399, 1411 (2011). Under the new “paradigm,” parties “have collaboratively built some of the most popular and important software technologies in the world, including Linux, Android, Apache Web Server, Firefox * * * and many others that power much of the Internet and computing world.” Asay, *Anticommons*, *supra*, at 283. That trend will only continue—especially as the world shifts to cloud computing, where 90% of workloads use the Linux open-source operating system and other open-source components. See Jonathan Corbet & Greg Kroah-Hartman, Linux Found., *2017 Linux Kernel Development Report 1* (Oct. 25, 2017), <https://www.linuxfoundation.org/publications/2017/10/2017-state-of-linux-kernel-development/>.

The proper paradigm is critical to fair-use analysis. Under the traditional single-producer model, many assumed that strong intellectual-property protection for

software was “the only feasible way to cover the costs of innovation.” Baldwin & von Hippel, *supra*, at 1411. But the collaborative ecosystem shows that the traditional calculus “that software creators will not incur the costs necessary to develop software without exclusive rights in that software” no longer holds. Asay, *Anticommons*, *supra*, at 271. Instead, “any given piece of software may include dozens, hundreds, or even thousands of copyright holders.” *Id.* at 279. With decreased collaboration costs and new revenue streams that do not depend on a proprietary model, companies are incentivized to innovate even if they do not capture monopoly profits. See Baldwin & von Hippel, *supra*, at 1399-1400.

Much of this collaborative process is facilitated by various copyright-based licensing agreements that have developed over time. Asay, *Anticommons*, *supra*, at 279. Nonetheless, collaborative innovation frequently occurs without the express permission of the copyright holder—yet within the boundaries of copyright law. As explained below (at 12-14), the computer industry has adapted in particular to the breathing room that courts have construed the fair use doctrine to provide. See Joseph Gratz & Mark A. Lemley, *Platforms and Interoperability in Oracle v. Google*, 31 Harv. J.L. & Tech. 603, 610 (2018). Thus, an impractically rigid approach to copyright could “make * * * collaboratively built resource[s] more difficult” to produce, stymieing the software-development models that are now ascendant. Asay, *Anticommons*, *supra*, at 268.

2. Software differs from traditional literary works in other critical respects. Unlike a novel, software is not purely a work of creative expression—it is a “hybrid” of creative and functional elements. *Altai*, 982 F.2d at 712. A programmer’s “program structure and design may be

highly creative and idiosyncratic.” *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992), as amended (Jan. 6, 1993). But other aspects of software are utilitarian and serve largely functional purposes, such as “facilitat[ing] communication between the user and the computer.” *Apple Comput., Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1444 (9th Cir. 1994). Software thus “‘hover[s] * * * closely to the elusive boundary’” between idea and expression that marks copyright’s bounds. *Lexmark Int’l, Inc. v. Static Control Components*, 387 F.3d 522, 535 (6th Cir. 2004). The software interfaces at issue in this case are one example. While they reflect certain minimal creativity, they are largely functional—they allow a computer programmer to invoke a function on a device. See Pet. App. 179a-180a.

Unlike a novel, moreover, software often is not a self-contained work that an end user consumes in isolation. Even in the earliest software fair-use cases, courts recognized the need for “compatib[ility]” and sharing across platforms and devices. *Sony Comput. Entm’t, Inc. v. Connectix Corp.*, 203 F.3d 596, 607 (9th Cir. 2000). In today’s “cloud and Internet of Things economies,” the computer industry relies even more on “shared resources and interoperability between heterogeneous computing devices and services.” Asay, *Anticommons*, *supra*, at 271. Third parties often reuse functional software code to create new products and to enable all sorts of software and hardware devices “to interoperate and share data.” *Id.* at 279.

Reuse of functional code has other advantages. It frees programmers to focus on the creative aspects of software development, such as coming up with ideas for new applications. Where certain functional code serves as an industry standard (de facto or otherwise) for

executing computer operations, that liberates programmers from having to rewrite the functional “plumbing” for each new program or device. See Asay, *Anticommons, supra*, at 304. Allowing third parties to reuse the functional aspects of existing code thus permits them to shift resources toward innovation rather than duplicating existing infrastructure. *Ibid.*

* * * * *

Software thus presents unique practical challenges for copyright law. Compared to traditional works, there are likely to be both more copyright holders in any given piece of software, and a greater practical need to reuse aspects of software to foster follow-on innovation. Those “changes” from the literary context require a reasoned “response” from the courts when applying fair use. *Sony*, 464 U.S. at 430. Until the decision below, courts largely have gotten the balance right, see pp. 9-12, *infra*, and the computer industry has thrived as a result, see pp. 12-14, *infra*. But the Federal Circuit’s uncritical application of copyright law to software as if it were a novel risks upsetting long-settled expectations and defeating copyright law’s purpose of “promot[ing] the Progress of Science and useful Arts.” U.S. Const. art. I, §8, cl. 8. It threatens to introduce uncertainty and transaction costs into an industry that has largely organized itself to avoid such constraints.

B. A Flexible Fair Use Doctrine Is Essential To Accommodate the Modern Software Development Approach—As Courts Have Long Recognized

Copyright promotes two fundamental but competing interests: On the one hand, it seeks to “assure[] authors the right to their original expression”; on the other, it “encourages others to build freely upon the ideas and

information conveyed by a work.” *Feist Publ’ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349-350 (1991). To that end, copyright provides broad protections for an author’s “creative expression,” which “falls within the core” of the work copyright law is intended to foster. *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 586 (1994). But an array of doctrines also makes clear that copyright law allows great latitude for the reuse of facts, ideas, and other functional elements underlying an author’s work. Those include the “idea/expression” dichotomy, *Feist*, 499 U.S. at 350, *scènes à faire*, *Swirsky v. Carey*, 376 F.3d 841, 850 (9th Cir. 2004), and most relevant here, fair use, *Campbell*, 510 U.S. at 576; see *Stewart v. Abend*, 495 U.S. 207, 237 (1990) (“[F]air use is more likely to be found in factual works than in fictional works.”).

Fair use is a “flexible” and adaptable doctrine. *Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146, 1163 (9th Cir. 2007). That extends to accounting for the technological realities the copyrighted work presents. See *Sony*, 464 U.S. at 430. For decades, courts have tailored fair-use analysis to account for software’s hybrid nature—affording greater protection to the creative aspects of software, while offering less protection to the functional and utilitarian aspects. Courts have been particularly willing to find fair use where functional aspects of software were used to achieve interoperability of software and devices.

In *Sega*, for example, the Ninth Circuit upheld Accolade’s copying of object code to develop video games that could be played on Sega’s Genesis console. 977 F.2d at 1514-1515, 1525. The code was “essentially utilitarian”—covering the “subroutines” that allowed “the user to interact with the video game” and “the game cartridge to

interact with the console”—and thus warranted only “thin” copyright protection. *Id.* at 1524-1525. Accolade copied it, moreover, not to appropriate expressive content, but to access “functional” elements needed for “compatibility.” *Id.* at 1522. And even though Accolade created “a *competing* product,” that commercial purpose did not “preclude[] a finding of fair use” because it was “rebutted by” the resulting “public benefit”: Accolade’s use led to an “increase in the number of independently designed video game programs offered for use with the Genesis console.” *Id.* at 1522-1523 (emphasis added).

The Ninth Circuit applied similar reasoning to reach a similar result in *Connectix*. 203 F.3d 596. Connectix had copied Sony’s “BIOS”—software that controlled the basic functions of Sony’s PlayStation game console. *Id.* at 603. Connectix did so in connection with creating new software that enabled users to play video games that had been developed for Sony’s console on PCs. *Id.* at 601. The court noted that Sony’s BIOS “lies at a distance from the core [of copyright protection] because it contains unprotected [functional] aspects.” *Id.* at 603. The BIOS code thus was entitled to a “lower degree of protection than more traditional literary works.” *Ibid.* The court also found that Connectix’s program was a transformative use of the BIOS, because it “afford[ed] opportunities for game play in new environments.” *Id.* at 606. The court held that Connectix’s copying was fair use, despite the fact that it was done to create a product that *competed* with Sony’s product. *Id.* at 608.

Similar examples abound. Time and again, courts have held that copying software to access its functional elements—to develop follow-on or interoperable technologies—is fair use that furthers the purposes of copyright law. See, e.g., *Lewis Galoob Toys, Inc. v. Nintendo*

of Am., Inc., 964 F.2d 965, 971 (9th Cir. 1992) (fair use for consumers to use a product that was compatible with Nintendo’s games); *Atari Games Corp. v. Nintendo of Am. Inc.*, 975 F.2d 832, 843-844 (Fed. Cir. 1992) (reverse-engineering a game console’s software to make compatible games was fair use).

Until the Federal Circuit’s decision below, the broad application of fair use in such circumstances was considered “settled law.” Gratz & Lemley, *supra*, at 610. The computer industry has structured its conduct in reliance on the breathing room for reuse of functional code that such decisions provided.

C. Experience Has Shown That a Flexible Fair Use Doctrine Fosters Innovation in Computer Software

Experience has shown that the pragmatic approach courts have long taken to fair use of software code has fostered the “growth in creative expression * * * that the Copyright Act was intended to promote.” *Sega*, 977 F.2d at 1523. Indeed, some have urged that copyright’s “solicitousness to copying for the purpose of interoperability is the reason we have a vibrant and competitive [computer] industry” today. Gratz & Lemley, *supra*, at 610.

The video-game industry, for instance, has flourished since the seminal decisions finding fair use in the 1990s. Previously, game development was tied to the maker of the game console and its licensees. Allowing third parties to develop compatible games by reverse-engineering game-console software “facilitat[ed] the entry” of “new competitor[s].” *Sega*, 977 F.2d at 1523. Since then, the video-game industry has grown to generate over \$36

billion in annual U.S. revenues.² Microsoft chose to enter the console market after *Sega* was decided, releasing its Xbox line. Microsoft has invested billions of dollars in creating its own Xbox games, but it and its customers have also benefited from third-party developers expanding the array of games that can be played on the Xbox.

Nor did the video-game industry’s growth come at the expense of the original copyright holders. In *Connectix*, Sony argued that it would lose sales and profits if Connectix were permitted to create a competing platform that could run games created for Sony’s PlayStation. 203 F.3d at 607. But the latest PlayStation has sold more than 86 million units.³ In one recent quarter, “Sony’s PlayStation arm brought in nearly \$750 million in profit,” making it the company’s most profitable department.⁴

Thus, a central premise of the Federal Circuit’s approach—that the jury’s fair-use finding here would undermine incentives to produce computer programs—is belied by history. A flexible application of fair use has not harmed the ability of software producers to enter into productive copyright-licensing arrangements and otherwise recoup their investments in innovation. Copyright holders have continued to thrive because allowing reasonable fair use of functional code enables innovation that

² *US Video Game Industry Revenue Reaches \$36 Billion in 2017*, Entm’t Software Ass’n (Jan. 18, 2018), <http://www.theesa.com/article/us-video-game-industry-revenue-reaches-36-billion-2017/>.

³ Jon Russell, *Sony Posts \$2.1B Profit as PlayStation Sales Keep on Growing*, TechCrunch (Oct. 30, 2018), <https://techcrunch.com/2018/10/30/sony-posts-2-1b-profit-as-playstation-sales-keep-on-growing>.

⁴ Jamie Rigg, *Sony Can’t Stop Making Money from PlayStation*, Engadget (July 31, 2018), <https://www.engadget.com/2018/07/31/sony-playstation-4-millions>.

serves the public and creates new opportunities for the whole market to grow.

II. THE FEDERAL CIRCUIT’S DECISION DEFIES SETTLED FAIR-USE PRINCIPLES AND MISAPPREHENDS THE NATURE OF THE COMPUTER INDUSTRY

The Federal Circuit’s decision here—finding that Google’s reuse of functional Java code was not fair use as a matter of law—upends the computer industry’s settled expectations about fair use of software code. Instead of treating fair use as a “flexible” doctrine that can adapt to address software’s dual nature, *Perfect 10*, 508 F.3d at 1163, the Federal Circuit took a rigid view that treats even the functional aspects of software as if they were entitled to the same protection as creative literary works. It also took a straitjacketed view of the “transformative use” factor of fair use, failing to acknowledge that Google’s reuse of the Java software interfaces in its Android operating system has made a world of new features and functions possible for Java programmers and consumers alike.

The purpose of copyright law is to “promote” “the Progress of Science and useful Arts” overall. U.S. Const. art. I, §8, cl. 8. The Federal Circuit’s cramped fair-use analysis defies that purpose, threatening to disrupt collaborative software development and restrict creativity in the most vital and inventive sector of our economy. This Court’s review is warranted.

A. The Federal Circuit’s Disregard of the Functional Nature of Oracle’s Declaring Code and SSOs in the Fair-Use Analysis Defies Precedent and Industry Reality

The Copyright Act requires consideration of the “nature of the copyrighted work” in any fair-use analysis.

17 U.S.C. § 107(2). That factor “calls for recognition that some works are closer to the core of intended copyright protection than others, with the consequence that fair use is more difficult to establish when the former works are copied.” *Campbell*, 510 U.S. at 586. Conversely, “[w]orks that are merely compilations of fact” or of “functional concepts” receive “‘thin’” protection, so fair use is easier to establish. *Sega*, 977 F.2d at 1524 (quoting *Feist*, 499 U.S. at 349). Given that software may contain both highly creative elements and essentially functional elements, courts have long recognized that a clear focus on “the nature of the copyrighted work” taken is “particularly significant” in software cases. *Micro Star v. Formgen Inc.*, 154 F.3d 1107, 1113 (9th Cir. 1998).

For example, in *Connectix*, the copied work consisted of utilitarian software code that controlled the basic functions of the PlayStation game console. 203 F.3d at 603. Because of its functional nature, the court explained, the code “lies at a distance from the core” of copyright protection, *ibid.*—a factor that “strongly favor[ed]” finding fair use, *id.* at 605. Similarly, in *Sega*, the Ninth Circuit emphasized that the functional nature of the object code that was copied was “important to the resolution” of the fair-use question. 977 F.2d at 1522.

In the decision below, the Federal Circuit took the opposite view. It declared that the nature of the copyrighted software at issue is “not * * * terribly significant in the overall fair use balancing.” Pet.App. 42a. The court thus took no real account of the fact that, while the Java declaring code and the SSOs meet the minimum creativity requirements for copyrightability, they are essentially functional—they are the means by which a programmer triggers a function on a device when writing software in the Java language. See, *e.g.*, Pet.App. 126a,

226a, 228a. Under longstanding software copyright principles, the functional nature of the code should have been an analytical pivot point favoring fair use. But the Federal Circuit discarded its significance altogether, and instead treated the relevant software code like a highly creative work within the core of copyright’s protection.

In holding that the nature of the copyrighted work is not significant, the Federal Circuit cited only cases involving traditional creative works, such as fictional books, artistic dolls and images, and television programming. Pet.App. 42a-43a (citing *Dr. Seuss Enters., L.P. v. Penguin Books USA, Inc.*, 109 F.3d 1394 (9th Cir. 1997) (*Dr. Seuss’s The Cat in the Hat*); *Mattel Inc. v. Walking Mountain Prods.*, 353 F.3d 792 (9th Cir. 2003) (Barbie doll); *Fox News Network, LLC v. TVEyes, Inc.*, 883 F.3d 169 (2d Cir. 2018) (news broadcasts), cert. denied, 139 S. Ct. 595 (2018)). It is unsurprising that courts may “give little attention to the nature-of-the-work factor in run-of-the-mill fair use analyses,” Pamela Samuelson & Clark D. Asay, *Saving Software’s Fair Use Future*, 31 Harv. J.L. & Tech. 535, 560 (2018), where the copied works contain the “‘creativity, imagination and originality’” at the heart of copyright protection, Pet.App. 42a (quoting *Dr. Seuss*, 109 F.3d at 1402). But software requires a different approach. As this Court has explained, copyright law must “respon[d] to significant changes in technology.” *Sony*, 464 U.S. at 430. And for decades, courts understood that the nature-of-the-work “factor carries greater weight” in this context “because of software’s functional nature.” Samuelson & Asay, *supra*, at 560.

The Federal Circuit justified its contrary conclusion on the grounds that “allowing this one factor to dictate a conclusion of fair use in all cases involving copying of software” would “negate” Congress’s declaration “that

software is copyrightable.” Pet.App. 43a. But recognizing that the functional code here is entitled to thinner protection would not dictate the outcome in “all cases” involving software. Different aspects of software lie on a spectrum, with more creative elements lying closer to “the core of intended copyright protection” than the code here. *Campbell*, 510 U.S. at 586. Courts have proved more than capable of drawing that distinction and tailoring the degree of fair-use protection to the nature of the software code in the cases before them. See, e.g., *Wall Data Inc. v. L.A. Cty. Sheriff’s Dep’t*, 447 F.3d 769, 780 (9th Cir. 2006) (concluding that the “nature of the copyrighted work weigh[ed] against a finding of fair use” for computer terminal emulation software); *Sega*, 977 F.2d at 1525 (distinguishing between functional and expressive aspects of video-game code). The Federal Circuit’s failure to do so here—and its indication that such distinctions are “not * * * significant” in future cases—upsets the computer industry’s long-settled expectations, with potentially disastrous consequences for innovation. See pp. 21-24, *infra*.

B. The Federal Circuit Fundamentally Misunderstood What Constitutes a “Transformative Use” of Software

Another critical factor in the fair-use analysis concerns “whether and to what extent the new work is ‘transformative.’” *Campbell*, 510 U.S. at 579. The “central purpose of [that] investigation” is to determine “whether the new work merely ‘supersedes the objects’ of the original creation, or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message.” *Ibid.* (citations and brackets omitted). The latter such works “lie at the heart of the fair use doctrine’s guarantee of breathing

space within the confines of copyright, and the more transformative the new work, the less will be the significance of other factors * * * that may weigh against a finding of fair use.” *Ibid.*

The Federal Circuit concluded that Google’s use of the Java software-interface code was not transformative because “the purpose of the API packages in Android is the same as the purpose of the packages in the Java platform”; “Google made no alteration to the expressive content or message of the copyrighted material”; and “smartphones were not a new context.” Pet.App. 31a-32a. That analysis misapprehends the purposes of copyright law and the nature of the code at issue—with critical consequences for future software cases.

1. The Federal Circuit took a rigid view of the “purpose” of Google’s reuse that ignores the realities of the computer industry. The court found that the fact that the Java declaring code and SSOs “serve the same function in both” Java and Google’s Android meant that Google’s use was not transformative. Pet.App. 33a. But while that code served the same broad “purpose” in both works—calling on a device to perform a specific function—the same could be said of any software code that is reused. Unlike literary works, software code serves not to enlighten or entertain, but “to carry out specific, preassigned computing functions.” Clark D. Asay, *Transformative Use in Software*, 70 Stan. L. Rev. Online 9, 14 (2017). As a consequence, “reuses of software will typically implicate the very same functions.” *Ibid.* The Federal Circuit’s analysis thus leads to an absurd result: It makes it *more* difficult to establish fair use for re-using functional software than for re-purposing aspects of a creative fictional work. See *id.* at 10. That does not merely turn copyright law on its head. It

“imperil[s] the “productive balance that fair use helps strike between copyright holders and follow-on software innovators.” *Ibid.*

2. The Federal Circuit also missed the point in focusing on the fact that Google did not alter the “expressive content or message of the copyrighted material” *itself*, Pet.App. 31a-32a—as opposed to acknowledging what Google did with that code in its Android operating system.

The transformative-use factor properly asks “whether the *new work* * * * adds something new, with a further purpose.” *Campbell*, 510 U.S. at 579 (emphasis added). Here, the jury reasonably could have found that Google’s reuse of the Java software interfaces was transformative because Google utilized the Java code in the context of a totally different software program, Android, that implemented the functions that the Java code invokes using totally different code. See Pet.App. 218a-219a. And unlike the Java platform, which “was developed to run on desktop computers and enterprise servers,” Pet.App. 216a-217a, Android “was designed specifically for mobile devices,” Pet.App. 196a, and thus “ha[d] to accommodate” factors like “limited memory and battery life, that did not apply to [the Java platform],” Pet. 25.

The Federal Circuit summarily dismissed the notion that Google’s use of the Java software interfaces in the Android mobile-phone platform was transformative simply because Java “was already being used in smartphones.” Pet.App. 35a. But that broad statement has little bearing on whether Google’s use was, in reality, a transformative use of the code. Whether or not “other smartphone manufacturers” had already licensed Java for use in mobile phones, *ibid.*, the fact is that Android “completely transformed the mobile computing industry

and powered innovation in the smartphone market,” Asay, *Anticommons, supra*, at 315.⁵ Indeed, without new platforms like Android, a single mobile operating system—Apple’s iOS—likely would have dominated the smartphone market. Cf. Pet.App. 219a (“Android-based mobile devices * * * now comprise a large share of the United States market.”).

Ultimately, Google was “not seeking to appropriate the advances” in the Java software interfaces, but “to give [Java programmers] an option to exploit their own prior investment in learning” the Java language. *Lotus*, 49 F.3d at 821 (Boudin, J., concurring). And Android opened up new possibilities to Java programmers, fostering the development of additional, compatible programs. Because Google “facilitate[d] greater compatibility and collaboration” among Java programmers “outside of strictly Sun/Oracle products,” its use “represents a different purpose than that of the original creation, and arguably one with greater societal potential.” Asay, *Anticommons, supra*, at 314-315. It cannot be that Google’s use was not transformative *as a matter of law*.

3. The Federal Circuit’s emphasis on Google not having “alter[ed] * * * the expressive content or message of the copyrighted material” itself, Pet.App. 31a-32a, is misplaced for another reason. The expressive content in, for example, the Java declaring code, lies in the names chosen to invoke various functions. See Pet.App. 150a. The Federal Circuit could identify no way in which Google altering the names of functions in the declaring

⁵ Moreover, while Oracle has a copyright in Java, the law “does not confer” copyright holders in software with “control over the market for devices” that may run that software. *Connectix*, 203 F.3d at 607.

code would serve copyright law’s purpose of “promot[ing] the Progress of Science and useful Arts.” U.S. Const. art. I, §8, cl. 8. Having more names for the same software functions does not enrich society. Quite the opposite—that is akin to having “every typewriter maker * * * scramble the [QWERTY] keyboard.” Pet.App. 104a. In short, the Federal Circuit’s analysis represents the type of thinking this Court has warned against: It seeks to “simplif[y]” the fair-use analysis with “bright-line rules,” rather than performing “case-by-case analysis” and application “in light of the purposes of copyright.” *Campbell*, 510 U.S. at 577-578. For that reason, too, review is warranted.

III. THE FEDERAL CIRCUIT’S RIGID, NARROW APPROACH TO FAIR USE THREATENS THE VIABILITY OF THE INTERCONNECTED SOFTWARE ECOSYSTEM

The Federal Circuit’s decision will have ramifications far beyond the dispute between Oracle and Google over the Java code in this case. While fair use is supposed to involve a “case-by-case analysis,” *Campbell*, 510 U.S. at 577-578, the Federal Circuit made clear that it intended the analytical framework it adopted to “guide resolution of [the fair-use] question in all future cases” involving software, Pet.App. 18a.⁶ The Federal Circuit’s failure to take a view of fair use that accounts for the real-world uses of functional software code thus may have profoundly negative consequences for innovation in the computer industry.

⁶ The Federal Circuit does not have exclusive appellate jurisdiction over copyright cases. However, it will often be possible for copyright holders to seek Federal Circuit jurisdiction by asserting patent claims—over which that court does have exclusive jurisdiction, 28 U.S.C. § 1295(a)(1)—along with their copyright claims.

The Federal Circuit’s decision threatens the models of “open collaborative innovation” that are now established “sources of innovative products, processes, and services” in the computer ecosystem. Baldwin & von Hippel, *supra*, at 1411. Under such models, a company’s software product often is not an end point, but instead a launching pad for further innovation. “[O]pen source software development” is a prominent example. *Id.* at 1401; see pp. 6-7, *supra*. Another, perhaps lesser-known example involves commercial users of software products modifying existing software to better serve their needs. Such user-based innovation has resulted in a significant amount of “commercially significant product and process development and modification in many fields.” Baldwin & von Hippel, *supra*, at 1400.

A cutting-edge example of such follow-on innovation involves artificial intelligence. In that space, companies are developing highly sophisticated, deep-learning systems, yet it is recognized that third parties may have ingenious new ideas for services that utilize such systems’ capabilities.⁷ But if companies can no longer assume that reuse of functional elements of the original software product for such purposes will be protected as fair use, that threatens to impede such follow-on, collaborative innovation at the most basic level.

The Federal Circuit’s decision also threatens another pillar of today’s computer ecosystem—seamless interoperability across software platforms and hardware devices. Consumers already expect to be able to take a

⁷ See, e.g., Mark Kaelin, *Microsoft Cognitive Services: Leading the AI Charge*, TechRepublic (May 22, 2017), <http://www.techrepublic.com/article/build-2017-ai-will-change-everything-and-microsoft-looks-to-lead-the-way/>.

photo on their Apple phone, save it onto Google’s cloud servers, and edit it on their Surface tablets. And the coming “Internet of Things”—in which everything from thermostats to traffic lights will access the Internet—will push the demands of interoperability to new extremes.⁸ Such interoperability is often made possible through the reuse of common functional code. See Gratz & Lemley, *supra*, at 609-613. Under prior law like *Sega* and *Connectix*, companies could take comfort that reusing such code for the purpose of achieving interoperability would be fair use. See pp. 9-12, *supra*. But the Federal Circuit’s decision upends those assumptions, creating uncertainty and disincentives to innovation.

Ultimately, the Federal Circuit’s decision means less collaboration, less interoperability, and less innovation for consumers—the opposite of the progress copyright law is intended to foster. By contrast, the jury’s finding of fair use has no detrimental effect on the ability of software producers to recoup their investment in software creation. Technological changes have reduced the costs of innovation, and it is no longer the case that producers always require decades of exclusive rights to profit from their software creations. See pp. 6-7, *supra*. In addition, industry experience in the wake of decisions like *Sega* and *Connectix* demonstrates that robust application of fair use tends to expand the overall market for the technology at issue, to the benefit of the original copyright holders. See pp. 12-14, *supra*.

⁸ See Karen Rose et al., Internet Soc’y, *The Internet of Things: An Overview* 8-9, 15 (2015), <https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf>.

This Court should grant review and restore the flexible approach to fair use that is essential to striking the correct balance between copyright protection and follow-on innovation.

CONCLUSION

The petition for a writ of certiorari should be granted.

Respectfully submitted.

LISA W. BOHL
MOLOLAMKEN LLP
300 N. LaSalle St.
Chicago, IL 60654
(312) 450-6700

JEFFREY A. LAMKEN
Counsel of Record
MICHAEL G. PATTILLO, JR.
WILLIAM J. COOPER
MOLOLAMKEN LLP
The Watergate, Suite 660
600 New Hampshire Ave., NW
Washington, D.C. 20037
(202) 556-2000
jlamken@mololamken.com

Counsel for Amicus Curiae

FEBRUARY 2019