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 *AMICI CURIAE* THE COMPUTER
& COMMUNICATIONS INDUSTRY
ASSOCIATION AND INTERNET ASSOCIATION
IN SUPPORT OF PETITIONER**

————

MATT SCHRUERS
ALI STERNBURG
COMPUTER & COMMUNICATIONS
  INDUSTRY ASSOCIATION
25 Massachusetts Ave. NW
Washington, D.C. 20001

JONATHAN BERROYA
INTERNET ASSOCIATION
660 N. Capitol Street NW
Washington, D.C. 20001

JONATHAN BAND
  Counsel of Record
JONATHAN BAND PLLC
21 Dupont Circle NW
Suite 800
Washington, D.C. 20036
(202) 296-5675
jband@policybandwidth.com

Counsel for Amici Curiae

January 13, 2020

TABLE OF CONTENTS

TABLE OF CONTENTS—Continued

iii
TABLE OF AUTHORITIES

TABLE OF AUTHORITIES—Continued

Page(s)

Page(s)

TABLE OF AUTHORITIES—Continued

TABLE OF AUTHORITIES—Continued

Page(s)

INTEREST OF *AMICI CURIAE*[1]

The Computer & Communications Industry Association ("CCIA") represents more than 20 companies of all sizes providing high technology products and services, including computer hardware and software, electronic commerce, telecommunications, and Internet products and services – companies that collectively generate more than $540 billion in annual revenues.[2]

Internet Association ("IA") represents over 40 of the world's leading internet companies. IA is the only trade association that exclusively represents leading global internet companies on matters of public policy. IA's mission is to foster innovation, promote economic growth, and empower people through the free and open internet.[3]

Computer programs are critical elements of all the products and services provided by the industries served by *amici*'s members, as well as industries as diverse as consumer electronics and automobiles. *Amici*'s members thus have a large stake in the rules of software copyright: effective intellectual property

---

[1] Pursuant to Supreme Court Rule 37.6, *amici* affirm that no counsel for a party authored this brief in whole or in part, that no party or counsel for a party made a monetary contribution intended to fund the preparation or submission of this brief, and that no person other than *amici*, their members, or their counsel made such a monetary contribution. Both parties consented in writing to the filing of this brief.

[2] A list of CCIA members is available at https://www.ccianet.org/members. Google is a CCIA member, and Oracle and Sun Microsystems were formerly members of CCIA, but none of these parties took any part in the preparation of this brief.

[3] A list of IA members is available at https://internetassociation.org/our-members/. Google is a member of IA. As noted above, Google took no part in the preparation of this brief.

protection encourages developers to create new applications, but the improper extension of copyright law to the elements that enable interoperability discourages innovation and inhibits competition in the technology industries.

SUMMARY OF ARGUMENT

Few intellectual property disputes have been worthier of Chief Justice Roberts' observation in *eBay* that "a page of history is worth a volume of logic," *eBay Inc. v. MercExchange*, *LLC*, 547 U.S. 388, 394 (2006) (Roberts, J., concurring) (citations omitted). The history of *amici*'s industries has been defined by the evolution of copyright rules governing interoperability. Without understanding how we arrived at these rules that enable today's interoperable world, one cannot fully appreciate how severely the Federal Circuit's decisions contradict decades of industry norms. This brief therefore describes how a series of judicial and legislative decisions around the world laid a foundation of interoperability that modern technology industries have relied upon for decades to compete and innovate.[4]

The computer industry in the 1960s and 1970s was fragmented into walled gardens, the largest of which was dominated by IBM. As copyright protection was extended to software, IBM and the other computer manufacturers sought to use copyright to prevent competition within their gardens. In particular, the

---

[4] This history is discussed in detail in two books co-authored by counsel of record on this brief. Jonathan Band & Masanobu Katoh, *Interfaces on Trial: Intellectual Property and Interoperability in the Global Software Industry* (1995), *available at* http://www.policybandwidth.com/interfaces-2-0 ("*Interfaces 1.0*"); and Jonathan Band & Masanobu Katoh, *Interfaces on Trial 2.0* (2011), *available at* http://mitpress.mit.edu/books/interfaces-trial-20 ("*Interfaces 2.0*").

dominant manufacturers argued that copyright protection extended to software interfaces, the means by which different program elements could interact.[5] The Java SE declarations at issue in this case are one such software interface.

The incumbent manufacturers' strategy was assisted by *dicta* in two decisions issued by the Third Circuit: *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983); and *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986). These opinions suggested that software interfaces fell within the scope of copyright protection. Fortunately, guided by this Court's decision in *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991), the Second Circuit in *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), rejected the Third Circuit's approach to software interfaces. Courts in other circuits promptly joined the Second Circuit in repudiating the Third Circuit.

Additionally, based on *Altai*'s approach to software interfaces, the Ninth Circuit in *Sega Enters., Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992), found that copyright did not prohibit the copying that occurred during the course of reverse engineering software for the purpose of identifying software interfaces. Other courts followed *Sega*'s lead. These decisions further underscored the unprotectability of software interfaces.

As a result of *Altai*, *Sega*, and their progeny, by the mid-1990s, there was a broad consensus in the U.S. software industry that copyright did not protect software interfaces such as the declarations at issue here;

---

[5] The brief uses the term "software interface" in the same manner as Google in its opening brief: "in the ordinary sense of a means of connecting to, interacting with, or operating computer software." Google Br. at 5 n.2.

and that the reverse engineering necessary to identify interfaces did not infringe copyright. This consensus subsequently was reinforced by Congressional enactment of an interoperability exception in the Digital Millennium Copyright Act ("DMCA") in 1998, 17 U.S.C. § 1201(f), and adoption of free trade agreements containing similar exceptions. This consensus led to a two-decade period of explosive growth and innovation in the industry, characterized by the development of software that interoperated with software created by other companies. The walled gardens of the 1960s and 1970s were replaced by networks with products made by different vendors exchanging data locally and around the globe via the Internet.

The Federal Circuit's decisions below shatter this consensus. The Federal Circuit's 2014 decision relied on the Third Circuit's discredited analysis in *Franklin* to conclude that copyright extended to the software interfaces at issue in this case. *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1375 (Fed. Cir. 2014). The Federal Circuit's 2018 decision found that fair use did not allow the replication of software interfaces in competing products. *Oracle Am., Inc. v. Google LLC*, 886 F.3d 1179 (Fed. Cir. 2018). Affirmance of these decisions would threaten our interoperable world.

The Federal Circuit's decisions also run directly contrary to legal norms promoting software industry competition that have been adopted by more than 40 of our trading partners, including all members of the European Union, and jurisdictions around the Pacific and across the world. These norms developed in parallel with and based upon *Altai*, *Sega*, and Congressional enactment of Section 1201(f) of the DMCA.

In 1991, the European Union adopted the Software Directive, which expressly permitted software reverse

engineering for the purpose of developing interoperable products. Council of Ministers Directive 91/250/EEC of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J. (L 122). The Directive has been implemented by all EU member states. In 2012, the Court of Justice of the European Union ruled that software interfaces fall outside the scope of copyright protection. Jurisdictions in the Pacific Rim and elsewhere similarly have enacted copyright exceptions encouraging competition, following either the U.S. or the EU model.

Allowing the Federal Circuit's decisions in this case to stand would result in two inconsistent legal regimes for software interfaces. In the United States, software interfaces would fall within the scope of copyright protection. In the rest of the world, a competitor could reuse software interfaces without authorization. This inconsistency would provide both startups and established companies in other countries with a competitive advantage over firms in the United States. Over time, this advantage may force U.S. firms to move their research, testing, and distribution operations overseas to avoid paying license fees. This would threaten the vitality of the U.S. technology industries.

Before proceeding to the Argument, *amici* must first address the relationship of this case to interoperability. Oracle, the Solicitor General, and the Federal Circuit have summarily dismissed concerns about the impact of this case on interoperability, asserting that Android and Java SE are not interoperable. While Android and Java SE are not completely interoperable, they are as interoperable as possible given that they function in radically different environments. As the district court found, and the Federal Circuit incorrectly rejected, in order for at least some of the millions

of lines of preexisting Java code *"to run on Android, Google was required to provide the same java.package.Class.method( ) command system using the same names with the same 'taxonomy' and with the same functional specifications.* Google replicated what was necessary to achieve a degree of interoperability—but no more, taking care, as said before, to provide its own implementations." *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 1000 (N.D. Cal. 2012) (emphasis in original). *See* JA202 (Google witness Professor Astrachan testifying that "if the method declarations changed, then the software that had already been written would no longer work").

Moreover, the Java SE declarations at issue in this case are a form of software interface. They enable an application to invoke pre-written subroutines. As such, they provide the means by which different program elements can interact. If the declarations are subject to copyright protection, then other software interfaces are as well. The Java SE declarations cannot be distinguished from other software interfaces in a manner that makes sense from a computer science perspective. Accordingly, affirmance of the decisions below would have a profound negative impact on interoperability.

ARGUMENT

I. **The Federal Circuit's Decisions Threaten to Undermine the Pro-Interoperability Consensus in U.S. Copyright Law.**

A. **The Second Circuit in *Altai* Rejected the Third Circuit's Erroneous *Franklin* and *Whelan Dicta* Suggesting That Software Interfaces Fall Within the Scope of Copyright Protection.**

The origins of this case lie in the tumultuous changes in the computer industry in the 1980s. In the previous decades, computing environments were highly balkanized. Once a customer purchased a computer system, the customer was essentially tied to that system: the system was incompatible with products manufactured by other companies, and conversion costs were high. The customers were large enterprises or governments, and IBM dwarfed other manufacturers. During the 1960s and 1970s, IBM controlled as much as two thirds of the global mainframe market. *Interfaces 1.0* at 20.

When IBM introduced its highly successful System/360 mainframes in the 1960s, it believed it would derive its profits from the sale of its hardware. It literally gave the operating system software away as a means of increasing demand for its hardware. Additionally, IBM distributed its operating system in source code format to make it easier for customers and other firms to write compatible applications, which further reinforced the dominance of the IBM mainframes. *Id.* at 24-25.

During the 1970s, however, IBM began to change its strategy with respect to software. In large measure, this change was driven by the extension of copyright

protection to software.[6] In 1978, IBM for the first time attached copyright notices to the new releases of its operating system software, and began charging for its use. In 1983, it stopped distributing its operating system software in source code. By license, it prohibited the reverse engineering of the operating system object code and the reuse of programming materials. *Id*. at 25.

Through these measures, IBM sought to control compatible software development. It refused to provide other firms with the interface information necessary to achieve interoperability; it argued that software reverse engineering constituted copyright infringement; and it contended that the unauthorized use of its software interfaces also constituted copyright infringement. *Id*.

During this period, software's contribution to IBM's revenue base expanded significantly. Before 1969, IBM received virtually no income from software. By 1984, IBM's software revenue exceeded $12 billion. *Id*.

IBM began to enforce its new software copyright policy through infringement actions. In 1982, IBM sued NCR for allegedly infringing IBM's copyright in its Network Control Program ("NCP"). IBM initially supplied NCP to its customers free of charge, and NCR developed NCP-compatible software. IBM then asserted copyright in its new releases of NCP. In order to remain NCP-compatible, NCR had to reverse engineer NCP and replicate NCP's interfaces. After two years of

---

[6] Although the Copyright Office began registering computer programs in 1964, the copyrightability of software remained uncertain until the enactment of the Computer Software Protection Act in 1980. See *Interfaces 1.0* at 71, 77.

litigation, the parties settled on undisclosed terms. *Id*. at 26.

Similarly, in 1985, IBM initiated an arbitration proceeding against the Japanese computer company Fujitsu. IBM alleged that when developing IBM-compatible mainframes, Fujitsu infringed copyright in the IBM operating system. The arbitration resulted in Fujitsu paying IBM for the right to develop interoperable products. *Id*. at 27.

Other manufacturers followed IBM's lead, including in the emerging personal computer ("PC") market. This new, restrictive approach to software alarmed many industry participants. There was broad acceptance of the extension of copyright protection to software, and the treatment of the wholesale copying of source or object code as copyright infringement. Likewise, there was little controversy about detailed program structure falling within the scope of copyright protection, to the extent that the structure did not affect complementary products. But these new software copyright policies went much further than that. IBM and the other manufacturers sought to use copyright to control software interfaces, thereby controlling the terms of competition in the industry. Copyright protection for interfaces would allow a manufacturer to determine whether another firm could develop products that could operate in the manufacturer's computing environment. It also would enable the manufacturer to restrict the use of familiar interfaces in new environments, thereby impeding innovation and the movement of programmers from one platform to another.

The incumbent manufacturers' position that software interfaces fell within the scope of copyright protection was assisted by *dicta* in two Third Circuit copyright

decisions. First, in *Apple v. Franklin*, the court asserted that compatibility is "a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expression have merged." *Franklin*, 714 F.2d at 1253. Under this reasoning, copyright could protect software interfaces even if they were necessary to achieve interoperability—even if there was only one way to write a compatible program.[7]

Then, in 1986, in *Whelan v. Jaslow*, the Third Circuit further suggested that copyright protected all aspects of a computer program other than its basic purpose: "the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea."[8] *Whelan*, 797 F.2d at 1236. Driving this understanding was the court's belief that such broad copyright protection was necessary to provide firms with the incentive to invest in the development of software: "the rule proposed here . . . would provide the proper incentive for programmers by protecting their most valuable efforts." *Id.* at 1231.

The *dicta* in these two opinions took on lives of their own, and cast long shadows over the development of competitive products. *Interfaces 1.0* at 99. To be sure, courts and legal scholars disagreed with the *Franklin* and *Whelan dicta*. *See, e.g.*, *Plains Cotton Co-op Ass'n*

---

[7] Because Franklin conceded that it could have achieved interoperability without copying the Apple II operating system, it did not have a valid merger defense and the court therefore had no need to consider the impact of interoperability on merger.

[8] The *Whelan* court identified the idea in the case before it as "the efficient management of a dental laboratory," and extended protection to the structure and logic of the program, as well as its code. 797 F.2d at 1236 n.28.

v. Goodpasture Computer Serv., Inc., 807 F.2d 1256, 1262 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987) (expressly "declin[ing] to embrace *Whelan*"); and Donald S. Chisum et al., *LaST Frontier Conference Report on Copyright Protection of Computer Software*, 30 Jurimetrics J. 15 (1989) (leading copyright scholars rejecting the overprotection granted by *Franklin* and *Whelan*). Nonetheless, *Franklin* and *Whelan* engendered sufficient Fear, Uncertainty, and Doubt—"FUD"—about the lawfulness of developing interoperable products that firms were reluctant to do so.[9] *Interfaces 1.0* at 99.

The critical turning point in the development of a pro-competitive approach to software copyright law occurred in 1992 when the Second Circuit in *Computer Associates v. Altai, Inc.* rejected the *Franklin* and *Whelan dicta*. Informed by this Court's holding the year before in *Feist v. Rural Telephone* that the scope of copyright protection in utilitarian works is "thin" and that substantial effort cannot confer copyright protection on unprotectable elements, the Second Circuit found that "*Feist* implicitly undercuts the *Whelan* [incentive based] rationale," *Altai*, 982 F.2d at 711, and ruled that under Section 102(b), copyright did not extend to software interfaces. Other circuits promptly followed *Altai* and rejected *Franklin* and *Whelan*. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832 (Fed. Cir. 1992); *Sega*, 977 F.2d at 1525; and *Gates Rubber Co. v. Bando Chemical Indus., Ltd.*, 9 F.3d 823 (10th Cir. 1993).

In 1995, it appeared that the Court would resolve the split between the Third Circuit and the other

---

[9] The FUD acronym was popularized at the time to describe disinformation campaigns waged by dominant firms against interoperable developers.

Circuits on the scope of copyright protection for software generally and the protectability of software interfaces in particular when the Court granted cert. in *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807 (1st Cir. 1995). Because of Justice Stevens's recusal, however, the First Circuit's decision was affirmed only by an evenly divided court, *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 516 U.S. 233 (1996).

In the absence of definitive resolution by the Court, most lower courts continued to follow *Altai* and reject *Franklin* and *Whelan*. *See Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366 (10th Cir. 1997); *Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 534 (6th Cir. 2004). As Professor Paul Goldstein observed, "most courts of appeal that have subsequently addressed the question of scope of copyright protection for computer programs have, like *Computer Associates*, effectively rejected the *Whelan* approach." Paul Goldstein, 1 *Goldstein on Copyright* § 2.15.1 (2d ed. 2005). The Third Circuit, conversely, continued to rely upon *Franklin* and *Whelan* in *Dun & Bradstreet Software Servs., Inc. v. Grace Consulting, Inc.*, 307 F.3d 197, 216 (3d Cir. 2002).

This consensus (outside of the Third Circuit) that copyright should not inhibit competition did not arise in a vacuum. *Franklin* and *Whelan* aroused concerns in the computer industry that large incumbents might use copyright anticompetitively to restrict access to software interfaces and impede software compatibility. In December 1991, new entrants in the hardware and software sectors convened at the headquarters of Sun Microsystems—the company that wrote the Java SE libraries and thus the software interfaces at issue in this case—to discuss these concerns. Chaired by Sun's Deputy General Counsel Peter Choy, this

group—the American Committee for Interoperable Systems ("ACIS")—agreed upon a Statement of Principles declaring that "[t]he 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." ACIS, Statement of Principles (1991), *available at* CCIA, *Interoperability Resources*, https://www.ccianet.org/interop.

ACIS subsequently filed *amicus* briefs in support of interoperability in many of the cases cited above.[10] Respondent Oracle endorsed these principles when it joined ACIS. And both Oracle and Sun were members of *amicus* CCIA, which joined ACIS in some of these briefs.[11]

The incumbent companies fought vigorously against ACIS, filing *amicus* briefs supporting copyright

---

[10] ACIS filed *amicus* briefs in *Altai*; *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435 (9th Cir. 1994); *Bando*; *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532 (11th Cir. 1996); *Borland*; *DVD Copy Control Assoc. v. Bunner*, 113 Cal. Rptr. 2d 388 (Cal. Ct. App. 2001); *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335 (5th Cir. 1994); *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996); *Sega*; *Sony Computer Entm't, Inc. v. Connectix Corp.,* 203 F.3d 596 (9th Cir. 2000); and *Unix Sys. Lab., Inc. v. Berkeley Software, Inc.*, 832 F. Supp. 790 (D.N.J. 1993). These briefs are available at https://www.ccianet.org/interop.

[11] CCIA joined ACIS in briefs in *Borland, Bunner, and Connectix*. CCIA filed its own briefs supporting interoperability in *Bowers v. Baystate Tech., Inc.*, 320 F.3d 1317 (Fed. Cir. 2003); *Chamberlain Group v. Skylink Tech., Inc.*, 381 F.3d 1178 (Fed. Cir. 2004); *Lexmark, Inc.*; *Pulse Commc'ns, Inc. v. DSC Commc'ns Corp.*, 528 U.S. 923 (1999); and *Sega*. These briefs are available at https://www.ccianet.org/interop.

protection for software interfaces.[12] The courts generally adopted the positions advocated by ACIS and its allies, including Oracle.

B. The Ninth Circuit in *Sega* Held That the Copying Incidental to Software Reverse Engineering Does Not Infringe Copyright.

1. The Courts Permitted Software Reverse Engineering for the Purpose of Identifying Software Interfaces.

At the same time that U.S. courts were rejecting the *Franklin* and *Whelan dicta*, they also adopted a procompetitive approach to the permissibility of software reverse engineering. Because a program's interfaces usually are not readily apparent, developers seeking to interoperate often must reverse engineer the original program in order to identify its interfaces. Reverse engineering is a basic tool of software product development without which interoperability can be difficult or impossible to achieve. *See Interfaces 1.0* at 167-68.

The Court has long recognized that there is nothing wrong with studying a competitor's product to understand how it works and to figure out how to make a better product. Thus, in *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 476 (1974), the Court stated that "trade secret law . . . does not offer protection against discovery by fair and honest means, such as . . . by so-called reverse engineering, that is by starting with a known product and working backward to divine the process which aided in its development or

---

[12] Many of these briefs were filed by the Computer Business Equipment Manufacturers Association ("CBEMA"). *See generally Interfaces 1.0*, 99-101 (recounting parties' general arguments).

manufacture." The Court has recognized that "[r]everse engineering . . . often leads to significant advances in technology." *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 160 (1989).[13]

Copyright law, however, could impede software reverse engineering; because of software's nature, reverse engineering a program almost always requires making a reproduction or derivative work. Disassembly, for example, involves translating the machine-readable object code into human-readable source code. In the 1980s, it was unclear whether such reverse engineering techniques infringed copyright.

After the enormous success of the IBM PC in the early 1980s, other firms sought to develop IBM-compatible PCs that could run the applications designed for the IBM PC. The key to this compatibility was the IBM PC's Basic Input Output Operating System ("BIOS"). It was clear that copying the IBM PC BIOS wholesale would constitute infringement. Competitors, therefore, needed a non-infringing compatible BIOS. In the mid-1980s, Phoenix Technologies decided to try to develop such a BIOS. To do so, Phoenix engineers reverse engineered the IBM BIOS and described its operation in minute detail. Then, Phoenix brought a second team of programmers with no prior knowledge of the IBM BIOS and tasked it with writing new code based on the first team's functional specifications. The resulting Phoenix BIOS was different from the IBM code, but operated identically. *See Interfaces 2.0* at 60; U.S. Copyright Office, *Software-Enabled Consumer*

---

[13] In the Defend Trade Secrets Act, Congress provided that misappropriation of a trade secret by improper means "does not include reverse engineering, independent derivation, or any other lawful means of acquisition." 18 U.S.C. § 1839(b)(6).

Products 53 (Dec. 2016). To the surprise of industry participants, IBM did not sue Phoenix for copyright infringement. Nonetheless, firms desiring to manufacture IBM-compatible PCs refused to purchase the Phoenix BIOS until Phoenix obtained a significant insurance policy that enabled it to indemnify the manufacturers in the event they were sued by IBM. Phoenix began selling its BIOS to IBM competitors such as Hewlett Packard, Compaq, Dell, and Gateway. The competition among manufacturers caused PC prices to fall and sales to surge, benefiting consumers greatly. *Interfaces 1.0* at 31.

In the video game industry, Accolade was not as lucky as Phoenix in avoiding litigation.[14] By the early 1990s, two Japanese companies, Sega and Nintendo, dominated the global home video game market. Each developed a set of interfaces between their consoles and the cartridges containing their games, rendering the consoles incompatible with non-conforming cartridges. Because consoles were expensive, buyers effectively committed themselves to one environment; they could only purchase games compatible with their console. An independent game developer, therefore, could compete only if it could achieve compatibility with either Nintendo or Sega consoles. *Interfaces 1.0* at 184.

Nintendo and Sega would license their interfaces for a stiff license fee, with severe restrictions. Accolade, a small American video game developer, decided that Sega's fees and conditions were unreasonable. Accordingly, it sought to achieve compatibility via disassembling Sega's code to identify the interfaces,

---

[14] Video games are computer programs.

then writing its own code to implement those interfaces. *Id*.

Soon after Accolade released a Sega-compatible game cartridge, Sega sued it for copyright infringement. The district court granted Sega's motion for preliminary injunction, rejecting Accolade's argument that its disassembly of Sega's program was a fair use. The Ninth Circuit reversed, finding that "in light of the public policies underlying the [Copyright] Act, disassembly of a computer program in order to gain an understanding of the unprotected functional elements of the program was a fair use when the person seeking the understanding has a legitimate reason for doing so and when no other means of access to the unprotected elements exists." *Sega,* 977 F.2d at 1514.[15]

Numerous other courts agreed that the reproduction that occurs while reverse engineering in order to identify software interfaces is permitted fair use. *See, e.g.*, *Atari*; *DSC Commc'ns Corp. v. DGI Techs., Inc.*, 898 F. Supp. 1183 (N.D. Tex. 1995), *aff'd*, 81 F.3d 597 (5th Cir. 1996); *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532 (11th Cir. 1996); *DSC Commc'ns Corp. v. Pulse Commc'ns, Inc.*, 976 F. Supp. 359 (E.D. Va. 1997), *aff'd in part, rev'd in part, and vacated in part*, 170 F.3d 1354 (Fed. Cir. 1999); and *Sony Computer Entm't v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000). Other courts treated infringement actions to prevent reverse engineering as copyright misuse. *See, e.g.*, *Alcatel*

---

[15] ACIS filed an *amicus* brief in support of Accolade. The ACIS Statement of Principles stated that copyright does not "restrict the ability of others to reproduce all or part of a lawfully obtained program as a step in the development of competing products." *See* https://www.ccianet.org/interop. CBEMA filed an *amicus* brief in support of Sega.

U.S.A., Inc. v. DGI Techs., Inc., 166 F.3d 772 (5th Cir. 1999). *See also Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 261 (5th Cir. 1988) (permitting reverse engineering under 17 U.S.C. § 117).

2. The Reverse Engineering Decisions Underscored the Unprotectability of Software Interfaces.

The reverse engineering decisions reinforced the principle that copyright does not protect software interfaces. The decisions were premised on the unprotectability of the software interfaces the reverse engineer sought to identify and reuse.

In *Sega*, for example, the Ninth Circuit excused the copying incidental to Accolade's reverse engineering of Sega's products because the purpose of the reverse engineering was to uncover the unprotectable software interfaces necessary to achieve compatibility. The *Sega* court found that Accolade reverse engineered "Sega's software solely to discover the functional requirements for compatibility with the Genesis console—aspects of Sega's programs that are not protected by copyright. 17 U.S.C. § 102(b)." *Sega,* 977 F.2d at 1522. The Ninth Circuit explained that if reverse engineering were not permitted,

> the owner of the copyright gains a de facto monopoly over the functional aspects of his work—aspects that were expressly denied copyright protection by Congress. 17 U.S.C. §102(b). In order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator must satisfy the more stringent standards imposed by the patent laws.

Id. at 1526. Unfortunately, the Federal Circuit below misunderstood *Sega*'s teaching on this point. Because *Sega* was a fair use case, the Federal Circuit concluded that interoperability was relevant only to the question of the applicability of the fair use defense. But *Sega*'s repeated citation to the Copyright Act's subject matter restriction in Section 102(b) leaves no doubt that the Ninth Circuit believed that elements necessary for interoperability were unprotectable under Section 102(b). The Ninth Circuit's subsequent decision in *Connectix* was likewise predicated on the unprotectability of interfaces. 203 F.3d at 603.

C. *Altai* and *Sega* Brought Two Decades of Stability, Competition and Innovation.

By the mid-1990s, it was widely accepted that software interfaces fell outside the scope of copyright protection;[16] and that the copying incidental to the reverse engineering to identify those interfaces did not infringe copyright. *Interfaces 2.0* at 2-3. This consensus encouraged incumbent firms to adopt more flexible licensing policies. Because a competitor could lawfully uncover and reuse the software interfaces anyway,

---

[16] Indeed, the U.S. Department of Justice took this position in litigation. In one pleading, the Justice Department observed that "it is by now well established that the copyright in a computer program cannot extend to the functional aspects of that computer program; to design choices dictated by necessity, cost, convenience, or consumer demand." *See Interfaces 2.0* at 66. Citing *Mitel*, the Department noted that the "interface specifications of a communications protocol are freely copiable because they are functional rather than expressive." *Id*. And citing *Altai* as well as *Stewart v. Abend*, 495 U.S. 207 (1990), and *Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151 (1975), the Department argued that copyright does not provide an unbounded property right but rather a limited power designed to encourage the creation of new authorship. *Interfaces 2.0* at 66.

incumbent firms decided it made business sense to license their interfaces on reasonable terms and derive some revenue from them.[17]

The practice became particularly widespread in the PC market. In its rush to develop the PC market in the early 1980s, IBM relied on partners such as Intel for the microprocessor and Microsoft for the operating system. IBM attempted to maintain overall control of the architecture through the BIOS, but as noted above, Phoenix reverse engineered the IBM BIOS and sold its version to IBM competitors such as HP and Dell. Intel and Microsoft were more than happy to sell their microprocessors and operating systems, respectively, to IBM's competitors. Moreover, Microsoft licensed the Windows interfaces to independent software developers (such as Lotus 1-2-3) so that they could create applications that ran on Windows. *Interfaces 1.0* at 30. Developments in network technology meant that a firm could link together an array of low-cost PCs instead of leasing an expensive mainframe. *Interfaces 1.0* at 34. The PCs, the peripherals, and much of the software could be purchased from different vendors at competitive prices. Additionally, these networks could include more powerful computers, such as the workstations developed by Sun Microsystems.

The unprotectability of software interfaces also accelerated the use of open systems such as UNIX and its progeny. *Interfaces 1.0* at 33. This further lowered the barriers to entry to new firms that made innovative products that could interact with other products.

---

[17] For the competitor, licensing the software interfaces also often made business sense. Licensing the interfaces could be less costly and time consuming than identifying them through reverse engineering. *See Interfaces 2.0* at 56-60.

The network effects and the public benefits were far-reaching.

> ### 1. The Digital Millennium Copyright Act Further Reinforced the Consensus that Software Interfaces Do Not Receive Copyright Protection.

Legislation enacted by Congress during this period reinforced the consensus that software interfaces fell outside the scope of copyright protection. Section 1201 of the Digital Millennium Copyright Act ("DMCA"), passed by Congress in 1998, restricts the development, distribution, and use of a technology that circumvents technological protection measures ("TPMs") that protect an author's copyrights. While the DMCA was pending before Congress, interoperable developers explained that the act of reverse engineering could require the circumvention of a TPM.[18] Moreover, the incorporation in competitive products of the interfaces learned through reverse engineering could run afoul of the DMCA's prohibition on the manufacture and distribution of circumvention technologies. This would particularly be the case when a company placed a software "lock" on a program that prevented access to the program, and the competitor circumvented that software lock to achieve interoperability. Thus, Section 1201 could have prevented a developer of

---

[18] In a 1998 press release, Michael Morris, then Vice President and General Counsel of Sun Microsystems, argued that the legislation would "impose[] a new and unnecessary layer of restraint on lawful access to those unprotected elements of computer programs that are necessary to achieve interoperability, thus placing developers of interoperable products at the mercy of proprietary vendors." Press Release, Sun Microsystems, House IP Subcommittee Action Threatens Internet Competition (Mar. 1, 1998).

interoperable products from exercising his fair use privileges recognized in *Sega* and its progeny.

In response to this advocacy, Congress included in the DMCA an exception explicitly directed at software reverse engineering for purposes of interoperability. Section 1201(f) specifically allows software developers to circumvent TPMs in a lawfully obtained computer program in order to identify the elements necessary to achieve interoperability of an independently created computer program with other programs.[19] Furthermore, a person may develop, distribute, and employ the means to circumvent TPMs for the purpose of achieving interoperability. 17 U.S.C. § 1201(f)(3).

The Senate Judiciary Committee report on the DMCA explained the policy underlying Section 1201(f) as being "intended to allow legitimate software developers to continue engaging in certain activities for the purpose of achieving interoperability to the extent permitted by law prior to the enactment of this chapter." S. Rep. No. 105-190 (1998), at 29. The Committee evidently understood that if a company placed on its program a TPM that prevented interoperability, a legal prohibition on circumventing that TPM could preclude other companies from developing products capable of operating in that company's computing environment. Citing *Sega*, the Committee stated that "[t]he objective is to ensure that the effect of current case law interpreting the Copyright Act is not changed by enactment of this legislation for

---

[19] Section 1201(f)(4) defines interoperability "as the ability of computer programs to exchange information, and of such programs mutually to use the information which has been exchanged." This definition, like much of the language of Section 1201(f), is derived from the EU Software Directive, discussed below in Section II.A.

certain acts of identification and analysis done in respect of computer programs." *Id.* The Committee concluded by noting that "[t]he purpose of this section is to foster competition and innovation in the computer and software industry." *Id.*

Additionally, the DMCA establishes a triennial rulemaking under which the Librarian of Congress, after receiving recommendations from the Register of Copyrights, is authorized to adopt exemptions to the DMCA's prohibition on the circumvention of TPMs when the prohibition would have an adverse effect on lawful uses of copyrighted works. The Register has repeatedly recommended, and the Librarian granted, exemptions predicted on the lawfulness of reusing software interfaces. *See, e.g.*, U.S. Copyright Office, *Section 1201 Rulemaking: Sixth Triennial Proceeding to Determine Exemptions to the Prohibition on Circumvention, Recommendation of the Register of Copyrights* 188 (Docket No. RM 2014-07, Oct. 8, 2015) (creation of a modified iPhone operating system for the purpose of enabling consumers to install unauthorized applications in their iPhones is fair use).

2. Free Trade Agreements Reflect the Consensus that Software Interfaces Are Outside the Scope of Copyright Protection.

Over the past fifteen years, the United States has negotiated free trade agreements ("FTAs") that included provisions modeled on the interoperability exception to Section 1201 of the DMCA. In addition to requiring parties to adopt prohibitions on the circumvention of TPMs, the FTAs permit countries to adopt exceptions for reverse engineering for the purpose of achieving interoperability. Thus, the recently concluded United

States-Mexico-Canada Agreement provides that each party may permit

> non-infringing reverse engineering activities with regard to a lawfully obtained copy of a computer program, carried out in good faith with respect to particular elements of that computer program that have not been readily available to the person engaged in those activities, for the sole purpose of achieving interoperability of an independently created computer program with other programs.

U.S.-Mexico-Canada Agreement, art. 20.66.4(a) (Dec. 10, 2019). Similar interoperability exceptions appear in the FTAs with Australia, Bahrain, Chile, Colombia, Costa Rica, Dominican Republic, El Salvador, Guatemala, Honduras, Korea, Morocco, Nicaragua, Oman, Panama, Peru, and Singapore. *See* Jonathan Band, *The Global API Copyright Conflict*, 31 Harv. J.L. & Tech. 615, 636 (2018) ("*Global API Conflict*"). As in the United States, many of these countries have adopted reverse engineering exceptions in their domestic law. *Id*.

D. The Federal Circuit's Decisions Shatter the Pro-Interoperability Consensus.

Over the past two decades, the technology industries have thrived in the stable legal environment that emerged after *Feist*, *Altai*, and *Sega*. The interoperability enabled by this legal regime has led to a dizzying pace of innovation and consumer choice. Unfortunately, the Federal Circuit's decisions in this case have disrupted this stable legal environment. The Federal Circuit's 2014 decision explicitly endorsed the long-discredited *dicta* in *Franklin* that compatibility is "a commercial and competitive objective which does not enter into the somewhat metaphysical issue of

whether particular ideas and expression have merged." 750 F.3d at 1375 (*quoting Franklin*, 714 F.2d at 1253).[20] By holding that software interfaces could be protectable under copyright, the Federal Circuit called into question a fundamental assumption of software development. The Federal Circuit compounded this error in its 2018 decision that Google's inclusion of Java SE declarations in the Android platform was not fair use. Affirmance of these two decisions would represent a major setback to competition and innovation in the U.S. technology industries.

II. Copyright Laws Around the World Promote Competition in the Technology Industries.

In addition to disrupting the stable copyright environment in the United States, the Federal Circuit's decisions run directly contrary to legal norms promoting competition in the technology industries that have been adopted by more than 40 of our trading partners, including all members of the European Union, and jurisdictions around the Pacific and across the world. These norms developed in parallel with and based upon U.S. judicial decisions such as *Altai* and *Sega* and Congressional enactment of Section 1201(f) of the DMCA. This parallel development is not surprising given the global nature of the technology industries.

A. European Union Law Mirrors the U.S. Pro-Interoperability Approach.

In 1991, the European Union adopted a Software Directive intended to harmonize the application of copyright to software in all the EU member states.

---

[20] In the Federal Circuit's decision, this language is no longer *dicta*.

While extending copyright protection to software, the EU decision makers sought to ensure that this legal protection would not prevent European companies from competing with U.S. companies such as IBM, Apple, and Microsoft. In particular, Article 6 of the Directive permits disassembly when "indispensably to obtain the information necessary to achieve . . . interoperability."[21] Article 5 further permitted other reverse engineering techniques for any purpose.[22] The Software Directive has been implemented by all member states of the EU, as well as Norway, Russia, Switzerland, and Turkey. *Interfaces 2.0* at 6.

While the Software Directive does not explicitly address the protectability of software interfaces, commentators generally perceived that "the law on software copyright interoperability issues seem[ed] quite settled on both sides of the Atlantic", and that software interfaces fell on the "idea" side of the idea/expression dichotomy. Pamela Samuelson, *The Past, Present, and Future of Software Copyright Interoperability Rules in the European Union and*

---

[21] The Directive uses the term "decompilation" to describe the same process the *Sega* court referred to as "disassembly:" the translation of machine-readable object code into human-readable source code. The legislative process leading to the adoption of Article 6 is discussed in detail in *Interfaces 1.0* at 227-41; and *Global API Conflict* at 617-19. The two industry factions that battled over software interfaces in the United States fought over the Directive as well, albeit in differently named coalitions: the European Committee for Interoperable Systems ("ECIS") and the Software Action Group for Europe. Sun and Oracle were members of ECIS.

[22] Underscoring the importance of these reverse engineering exceptions, the Directive established that any contractual provisions restricting them "shall be null and void." Software Directive, art. 9(2).

United States, 34(3) Eur. Intell. Prop. Rev. 229 (2010). The specific issue received scant attention until May 2012, when the EU's highest court, the Court of Justice of the European Union ("CJEU") ruled in *SAS Institute, Inc v. World Programming Ltd* [2012] 3 CMLR 4, ¶ 71, that program functionality, programming languages, and data formats were not protectable under the Software Directive.

The CJEU held that the Software Directive "must be interpreted as meaning that neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit its functions constitute a form of expression of that program and, as such, are not protected by copyright . . ." *Id*. ¶ 40. This affirmed World Programming's ability to create "middleware" that interoperated with SAS Institute's software. The CJEU observed that "the main advantage of protecting computer programs by copyright" as opposed to by patents "is that such protection covers only the individual expression of the work and thus leaves other authors the desired latitude to create similar or even identical programs," *id*. ¶ 41, provided that they refrain from copying protected expression. In other words, the CJEU reached precisely the same conclusion as the district court below, and the opposite of the Federal Circuit's 2014 decision.

B. Copyright Policies Around the Pacific and Across the World Align with U.S. and European Pro-Interoperability Law.

As policymakers in the Pacific Rim considered how best to encourage the development of domestic software industries, they followed either the U.S. fair use approach based on *Sega* or the specific statutory exception approach of the Software Directive—two

different means to the same end. *Global API Conflict* at 617.[23] After a decade-long copyright law review, Australia in 1999 followed the Directive model, adopting an exception for reverse engineering for purposes of interoperability. *Id.* at 631-33. Australian officials explained that "if Australian industry is to be allowed to compete on level terms with producers of similar products in the USA and Europe, Australian software copyright laws must be brought more into line with the law in these countries." Commonwealth, *Parliamentary Debates*, House of Representatives, 11 August 1999, 8479 (Daryl Williams, Attorney-General) (Austl.).

In the months before the 1997 turnover to China, the Hong Kong Legislative Council broadened Hong Kong's fair dealing provision to more closely resemble the fair use provision of the U.S. Copyright Act, in order "to encourage competition in the information technology industry by facilitating timely access to information and ideas underlying computer programs." Denise Yu, Sec'y of Trade and Indus., Speech by the Secretary of Trade and Industry on Resumption of Second Reading Debate 10 (June 24, 1997).

Similarly, Singapore in 1998 amended its fair dealing provision to "bring [it] in line with the United States, the United Kingdom, other European Union countries, Hong Kong, and Australia, which do not bar the use of copyright materials for commercial

---

[23] Foreign courts also relied upon *Altai*: *e.g.*, *Delrina Corp. v. Triolet Systems, Inc*., 9 B.L.R.2d 140 (Ont. Ct. of Justice 1993); *Matrox Elec. Sys., Ltd. v. Gaudreau*, [1993] R.J.Q. 2449 (C.S. Montreal); and *John Richardson Computers Ltd. v. Flanders and Chemtec Ltd*., 1993 FSR 497. *See Interfaces 1.0* at 147-50 and 262-69.

research." Second Reading of Copyright (Amendment) Bill of 1998 (Sing.) (Feb. 19, 1998).[24]

Over the following years, other Pacific Rim countries, including Canada, Chile, Japan, Malaysia, New Zealand, the Philippines, South Korea, and Taiwan all amended their copyright laws to encourage competition through interoperability, citing either the U.S. or the EU approach. *Global API Conflict* at 630, 634-35. Nations in other regions, including India, Israel, Kenya, Malawi, and Zimbabwe, have also explicitly embraced competition through interoperability in their copyright statutes. *Id*. at 634. They thus aligned themselves with the "hard-fought peace and harmony on interoperability issues" that had until recently prevailed in the United States and Europe. *See* Samuelson, *supra*.

III. Affirmance of the Decisions Below Will Greatly Disrupt the Global Technology Industries.

The Federal Circuit's decisions directly contradict a global consensus U.S. courts and Congress helped forge. Affirmance of the decisions below would result in two inconsistent legal frameworks for the reuse of software interfaces. In the United States, such reuse

---

[24] In the brief debate of the amendment in Parliament, Professor Chin Tet Yung said that it "is very important to ensure that there is a fair balance in any Copyright Bill between the interests of holders of rights in 'cutting edge' software and the interest of competitors who want to design and market non-infringing competing programmes which interface or are inter-operable with the basic programmes." Second Reading of Copyright (Amendment) Bill of 1998 (Sing.) (Feb. 19, 1998). In 2004, Singapore further amended its copyright law to include provisions modeled on reverse engineering provisions of the Software Directive. *Interfaces 2.0* at 167.

could occur only with the copyright owner's permission. In the rest of the world, the reuse would require no one's authorization. This inconsistency would place U.S. firms—particularly startups—at a competitive disadvantage to foreign firms.

Currently, with a uniform global software copyright system, startups in the United States have a head start over startups in other countries due to our academic computer science programs and our venture capital infrastructure. However, affirmance of the decisions below would result in our comparative advantage in these areas being overwhelmed by the more favorable software copyright framework in other countries. Startups in other jurisdictions, such as the UK, Israel, or Singapore could develop and test-market their products without paying license fees for the use of interfaces. When they finally had a product that succeeded in various markets around the world, they could afford to enter the U.S. market, where they would have to pay license fees for the interfaces. In contrast, startups located in the U.S. would have to pay license fees from day one on all the software they distributed anywhere in the world. These higher costs would make it significantly more difficult for them to succeed.

More established firms in the United States would also be at a disadvantage relative to foreign firms. The foreign firms would have to pay license fees for software interfaces only with respect to U.S. sales. Conversely, firms located in the United States would have to pay fees on both U.S. and foreign sales originating in the United States. Over time, this disadvantage may force U.S. firms to move more of their research, development, and distribution activities overseas. The technology industries that this country

pioneered would be driven offshore to jurisdictions with more favorable copyright regimes that we also helped pioneer.

Additionally, after affirmance, a rebalkanization of the U.S. technology industries could take place, particularly if incumbent firms refused to license their interfaces. Walled gardens of computing environments could reemerge, taking us back to the 1970s. This would decrease competition, thereby increasing prices and delaying innovation, to the detriment of consumers.

CONCLUSION

For the foregoing reasons, the Court should reverse the decisions below.

Respectfully submitted,

MATT SCHRUERS
ALI STERNBURG
COMPUTER & COMMUNICATIONS
   INDUSTRY ASSOCIATION
25 Massachusetts Ave. NW
Washington, D.C. 20001

JONATHAN BERROYA
INTERNET ASSOCIATION
660 N. Capitol Street NW
Washington, D.C. 20001

JONATHAN BAND
   Counsel of Record
JONATHAN BAND PLLC
21 Dupont Circle NW
Suite 800
Washington, D.C. 20036
(202) 296-5675
jband@policybandwidth.com

Counsel for Amici Curiae

January 13, 2020