

Nos. 2017-1118, 2017-1202

IN THE UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

ORACLE AMERICA, INC.,
Plaintiff-Appellant,

v.

GOOGLE INC.,
Defendant-Cross-Appellant.

On Appeal from the United States District Court for the Northern District of
California in Civil Action No. 3-10-CV-3561-WHA
Judge William Alsup

**BRIEF *AMICUS CURIAE* OF THE COMPUTER & COMMUNICATIONS
INDUSTRY ASSOCIATION IN SUPPORT OF GOOGLE INC.**

Matthew Schruers
Vice President, Law & Policy
Computer & Communications
Industry Association
655 15th Street NW, Suite 410
Washington, D.C. 20005
(202) 783-0070
mschruers@ccianet.org

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 Amicus Curiae

May 26, 2017

CERTIFICATE OF INTEREST

Pursuant to Rule 26.1 of the Federal Rules of Appellate Procedure and Federal Circuit Rule 47.4, Jonathan Band, counsel for *amicus curiae* the Computer & Communications Industry Association certifies the following:

1. The full name of every *amicus* represented by me is:

The Computer & Communications Industry Association.

2. The name of the real party in interest represented by me is:

The Computer & Communications Industry Association.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the *amicus curiae* represented by me are: None.

4. The names of all law firms and attorneys that appeared for the *amicus* now represented by me in the district court or are expected to appear in this court are:

The Computer & Communications Industry Association did not appear in the district court.

Before this court, the Computer & Communications Industry Association is represented by Jonathan Band PLLC, Jonathan Band, and Matthew Schruers.

May 26, 2017

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

TABLE OF CONTENTS

CERTIFICATE OF INTEREST	ii
TABLE OF CONTENTS	iii
TABLE OF AUTHORITIES	iv
INTEREST OF <i>AMICUS CURIAE</i>	1
INTRODUCTION AND SUMMARY OF ARGUMENT	2
ARGUMENT	8
I. Copyright Should Not Prevent the Portability of Programming Skills to Different Environments	8
A. Programmers Invest Significant Time and Resources Learning the Conventions of a Particular Programming Environment	9
B. Oracle Seeks to Appropriate Programmers’ Sunk Costs in Learning Java	12
C. Oracle’s <i>Amici</i> Misstate the Implications of this Case.....	14
II. Controlling Ninth Circuit Fair Use Precedent Supports a Fair Use Finding.....	15
A. Because the Second Fair Use Factor Requires Consideration of the Nature of the Work, Ninth Circuit Fair Use Decisions Concerning Computer Programs Are the Most Relevant Precedents	15
B. Google Had a Good Faith Belief that Replication of the Declaring Code Did Not Infringe Copyright	22
III. The District Court Excluded Evidence That Supported Google’s Fair Use Argument.....	25
CONCLUSION.....	29
CERTIFICATE OF COMPLIANCE.....	30
CERTIFICATE OF SERVICE.....	31

TABLE OF AUTHORITIES

Cases	Page(s)
<i>Apple Computer v. Microsoft</i> , 35 F.3d 1435 (9th Cir. 1994)	26
<i>Atari Games v. Nintendo of America</i> , 975 F.2d 832 (Fed. Cir. 1992)	23
<i>Bateman v. Mnemonics, Inc.</i> , 79 F.3d 1532 (11th Cir. 1996)	27
<i>Bikram’s Yoga College of India, L.P. v. Evolution Yoga, LLC</i> , 803 F.3d 1032 (9th Cir. 2015)	9
<i>Campbell v. Acuff Rose Music Inc.</i> , 510 U.S. 569 (1994)	15
<i>Chamberlain Group v. Skylink Techs.</i> , 381 F.3d 1178 (Fed. Cir. 2004)	20
<i>Computer Assocs. Int’l v. Altai, Inc.</i> , 982 F.2d 693 (2d Cir. 1992)	18, 19, 23, 26
<i>DVD Copy Control Assoc. v. Brunner</i> , 113 Cal. Rptr. 2d 388 (Cal. Ct. App. 2001)	27
<i>Engineering Dynamics v. Structural Sys.</i> , 26 F.3d 1335 (5th Cir. 1994)	27
<i>Gates v. Bando</i> , 9 F.3d 823 (10th Cir. 1993)	26
<i>Harper & Row Publ’rs v. Nation Enters.</i> , 471 U.S. 539 (1985)	15, 19
<i>L.A. News Serv. v. KCAL-TV Channel 9</i> , 108 F.3d 1119 (9th Cir. 1997)	15
<i>Lexmark Int’l v. Static Control Components, Inc.</i> , 387 F.3d 522 (6th Cir. 2004)	23
<i>Lorenzana et al. v. South American Restaurants Corp.</i> , 799 F.3d 31 (1st Cir. 2015)	9

<i>Lotus Dev. Corp. v. Borland Int'l</i> , 49 F.3d 807 (1st Cir. 1995)	<i>passim</i>
<i>Lotus Dev. Corp. v. Borland Int'l</i> , 516 U.S. 233 (1996)	5
<i>Mitel v. Iqtel</i> , 124 F.3d 1366 (10th Cir. 1997)	23
<i>Monge v. Maya Magazines, Inc.</i> , 688 F.3d 1164 (9th Cir. 2012)	15
<i>Oracle Am. v. Google</i> , 872 F. Supp. 2d 974 (N.D. Cal. 2012).....	7, 23, 24
<i>Oracle America, Inc., v. Google Inc.</i> , 750 F.3d 1339 (Fed. Cir. 2014)	16, 23, 24
<i>ProCD v. Zeidenberg</i> , 86 F.3d 1447 (7th Cir. 1996)	27
<i>Pulse Commc'ns v. DSC Commc'ns</i> , 528 U.S. 923 (1999).....	27
<i>Sega Enters., Ltd. v. Accolade, Inc.</i> , 977 F.2d 1510 (9th Cir. 1992)	<i>passim</i>
<i>Sony Computer Entm't v. Connectix Corp.</i> , 203 F.3d 596 (9th Cir. 2000)	<i>passim</i>
<i>Sony Corp. of Am. v. Universal City Studios</i> , 464 U.S. 417 (1984)	15
<i>Stewart v. Abend</i> , 495 U.S. 207 (1990)	19
<i>Unix Sys. v. Berkeley Software</i> , 832 F. Supp. 790 (D.N.J. 1993).....	26
<i>Wall Data Inc. v. L.A. Cty. Sheriff's Dep't</i> , 447 F.3d 769 (9th Cir. 2006)	22
<i>Wright v. Warner Books, Inc.</i> , 953 F.2d 731 (2d Cir. 1991)	15

STATUTES

17 U.S.C. § 107	3, 6
17 U.S.C. § 1201(f)	23, 27

17 U.S.C. § 1201(f)(2).....	2
-----------------------------	---

MISCELLANEOUS

American Committee for Interoperable Systems, <i>Statement of Principles</i> (1991).....	26
Jonathan Band, <i>Interfaces on Trial 3.0: Oracle America v. Google and Beyond</i> (2016)	24
Jonathan Band & Masanobu Katoh, <i>Interfaces on Trial 2.0</i> (2011)	27, 28
<i>Number of apps available in leading apps stores as of March 2017</i> , Statista.....	20
Pamela Samuelson, <i>Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement</i> , Berkeley Tech. L.J. (forthcoming)	24
Peter Norvig, <i>Teach Yourself Programming in Ten Years</i>	9
Peter Menell, <i>An Epitaph for Traditional Copyright Protection of Network Features of Computer Software</i> , 43 Antitrust Bulletin 651 (1998)	24
Peter Menell, <i>Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software</i> (UC Berkeley Pub. Law Research Paper No. 2893192, 2017).....	24
Press Release, Sun Microsystems, House IP Subcommittee Action Threatens Internet Competition (Mar. 1, 1998).....	27
Robert Kasunic, <i>Is That All There Is? Reflections on the Nature of the Second Fair Use Factor</i> , 31 COLUM. J.L. & ARTS 529 (2008).....	16
The White House, <i>Fact Sheet: A Year of Action Support Computer Science for All</i> , Dec. 5, 2016	11
The White House, <i>Fact Sheet: President Obama Launches New TechHire Initiative</i> , Mar. 9, 2015	10

Thomas Rogers & Andrew Szamosszegi, <i>Fair Use in the U.S. Economy</i> (2011)	8
U.S. Copyright Office, <i>Software-Enabled Computer Products</i> (2016)	21

INTEREST OF *AMICUS CURIAE*¹

The Computer & Communications Industry Association (“CCIA”) represents over twenty 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 CCIA members have a large stake in the rules of software copyright being properly designed: effective intellectual property protection encourages developers to create new applications, but the improper extension of copyright law to functional elements will discourage innovation and inhibit competition in the industry.

Four years ago, CCIA filed an *amicus* brief supporting affirmance of the district court’s order that the Java declaring code replicated by Google in Android fell outside the scope of copyright protection. Motivating this position was the belief that extending copyright protection to the information

¹ No counsel for any party authored this brief in whole or part, and no person other than *amicus curiae* or its counsel made a monetary contribution to the preparation or submission of this brief. Pursuant to Federal Rule of Appellate Procedure 29(a), all parties have consented 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 were formerly members of CCIA, but none of these parties took any part in the preparation of this brief.

necessary to achieve interoperability between programs would restrict CCIA members' ability to develop competitive products.

Although this Court found that Android and Java are not interoperable, this concern remains the same. If a firm sought to achieve interoperability with the Java (or Android) APIs, it would have to copy the Java (or Android) declaring code. And if copying information “for the purpose of enabling interoperability of an independently created computer programs with other programs,” *see* 17 U.S.C. § 1201(f)(2), infringes copyright, competition is impaired.

Competition also is impaired if copyright law prevents the replication of a subset of declaring code to make it easier for programmers to operate on a new platform. Restricting the portability of *programmers* can be just as anticompetitive as restricting the portability of *programs*. By enabling the mobility of programmers, the jury's fair use determination preserves the ability of CCIA members to develop innovative and competitive products.

INTRODUCTION AND SUMMARY OF ARGUMENT

In its opening brief, Oracle completely ignores the Ninth Circuit precedents most relevant to this case: *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1993), and *Sony Computer Entertainment, Inc., v. Connectix Corporation*, 203 F.3d 596 (9th Cir. 2000). In both *Sega* and

Connectix, the courts ruled that the fair use right, 17 U.S.C. § 107, permitted the copying of expression for the purpose of developing interoperable products with similar functionality to the plaintiffs' products. In both cases, the court sought to alleviate the harm to consumers resulting from being "locked-in" to a particular computing environment. In *Sega*, the court permitted the copying of expression for the purpose of developing new games that consumers could operate on the Sega platforms they had previously purchased; and in *Connectix*, the court permitted the copying of expression for the purpose of allowing consumers to operate their Sony games on a new platform—their personal computers. In both cases, the courts employed the fair use doctrine to ensure that copyright law did not render consumers captives of a particular vendor due to the consumers' prior investment in that vendor's products.

This case involves a different sort of interoperability. Instead of the interoperability between two computer programs, this case involves the interoperability of programming skills: the extent to which programmers can transfer their skills from one programming environment to another.

Although the nature of the interoperability is different in this case from *Sega* and *Connectix*, the underlying policy concern of preventing copyright from unreasonably restraining competition is the same. Just as fair use prevented

consumer lock-in in *Sega* and *Connectix*, so too should fair use prevent programmer lock-in in this case.

Programmers are the immediate beneficiaries of this interoperability. If their skill sets were not transferrable, they would have to start learning from scratch every time they work in a new environment. Software firms also benefit from this interoperability. If programmers' skills were not portable, then firms would need to convince programmers to learn a new toolset to work in a new environment, leading to slower adoption and higher training costs. But consumers are the ultimate beneficiaries of the interoperability of skills, as higher training costs for programmers are passed on to them. Moreover, the proliferation of programming environments enabled by the portability of skills means more innovation, competition, and consumer choice. It results in the "growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote." *Sega*, 977 F.2d at 1523.

This brief first argues that copyright should not prevent the portability of programming skills to different environments. Programmers must invest significant time and resources to learn the conventions of a particular programming environment, meaning that there are significant switching

costs for a programmer to learn a new set of conventions. Moreover, there is a shortage of skilled programmers in the United States. As a result, a new entrant must attract programmers from other firms in order to participate in the market. The shortage of skilled programmers thus constitutes a significant barrier to entry in the software industry. This barrier becomes near insurmountable if the programmers must learn a new set of programming conventions to operate in the new entrant's programming environment. Accordingly, portability of programming skills is essential to competition in the software industry.

Oracle seeks to prevent this portability so that it can appropriate the value of Java programmers' enormous investment in learning how to program in the Java environment. However, as Judge Boudin explains in his concurring opinion in *Lotus Development Corp. v. Borland Int'l Inc.*, 49 F.3d 807 (1st Cir. 1995) (Boudin, J., concurring), *aff'd by an equally divided Court*, 516 U.S. 233 (1996), copyright law should not enable this appropriation. Limiting doctrines such as fair use exist precisely to prevent this appropriation.

Oracle's entertainment industry *amici* suggest that affirmance would allow the migration of their works to new platforms without their authorization. This completely misstates the implications of this case. The

jury did not permit the wholesale migration of the Java API's millions of lines of code to a smartphone platform. Rather, the jury permitted the replication of specific elements of the Java API—0.4% of J2SE, comprising 0.08% of Android code, *see* Google Br. at 17—so that Java programmers would be able to transfer their knowledge to a new environment.

This brief then explains that controlling Ninth Circuit precedent requires a fair use finding. The second fair use factor is “the nature of the copyrighted work.” 17 U.S.C. § 107(2). Yet, Oracle completely ignores the most relevant Ninth Circuit fair use decisions involving computer programs: *Sega* and *Connectix*. Moreover, Oracle and its *amici* overlook the highly functional nature of computer programs, and the appropriate role of fair use to prevent the anticompetitive results of rigid application of copyright to functional works. Under the reasoning of Oracle and its *amici*, no use of a functional work could ever be transformative because the new work invariably would have a similar function as the original work. However, the Ninth Circuit found the *Connectix* Virtual Play Station to be transformative, even though it performed the same function as the Sony software, because it performed this function on a different platform; it allowed users to play their Sony videogames on their personal computers. The jury here could reasonably have concluded that Android was similarly transformative

because it provided Java programmers with a new platform for which they could create applications.

The Ninth Circuit also recognizes a fifth fair use factor, whether the defendant acted in good faith. Google here acted in good faith because it had objectively reasonable grounds for believing that its conduct was noninfringing. The district court itself believed that the declaring code replicated by Google was outside the scope of copyright protection. *Oracle America, Inc. v. Google Inc.*, 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (“*Oracle I*”). Additionally, decisions prior to the development of Android, including *Sega*, *Connectix*, and *Borland*, could reasonably be interpreted to indicate that the declaring code was not protectable. Further, many articles published by legal scholars prior to the development of Android could reasonably be understood to indicate that the declaring code was not protectable.

Lastly, Oracle complains that the district court improperly excluded evidence about the harm Android may have caused to markets other than the mobile market. However, the district court also excluded evidence that would have benefited Google’s fair use claim. In particular, the district court excluded evidence that Sun Microsystems (the creator of the Java API) and Oracle had previously taken legal positions opposite to those they took in

this case. The district court judge excluded this evidence in an effort to control his courtroom and keep a technically complex case manageable for a jury of ordinary citizens. If Oracle is granted a new trial to allow evidence of the impact on the desktop market, however, then Google should be allowed to introduce evidence on Oracle's reversal of its legal position.

ARGUMENT

I. Copyright Should Not Prevent the Portability of Programming Skills to Different Environments.

As the U.S. economy has transformed into a services-focused “knowledge economy,” copyright regulates an increasingly large aspect of contemporary business. Oracle's *amici* argue that copyright is important to the economy. *See* BSA Br. at 6-8. What is equally important, however, is that copyright does not obstruct the wheels of lawful commerce. This is achieved by limitations and exceptions such as fair use. Copyright exceptions matter to large portions of the U.S. economy. Thomas Rogers & Andrew Szamoszegi, *Fair Use in the U.S. Economy* (2011), <http://www.ccianet.org/wp-content/uploads/library/CCIA-FairUseintheUSEconomy-2011.pdf>.

The industries represented by Oracle's *amici* themselves admit that they frequently depend on fair use in their own activities. MPAA Br. at 1 (MPAA members “regularly rely on the fair use defense in producing and

distributing their expressive works”); Copyright Alliance Br. at 2; Oman Br. at 2-3. Beyond Oracle’s *amici*, numerous other professions depend on meaningful boundaries to copyright protection, from chefs, *Lorenzana et al. v. South American Restaurants Corp.*, 799 F.3d 31 (1st Cir. 2015), to yogis, *Bikram’s Yoga College of India, L.P. v. Evolution Yoga, LLC*, 803 F.3d 1032 (9th Cir. 2015). Software programmers and their employers also depend on meaningful limitations to copyright if programmers’ skills are to be portable from one environment to another. This portability is critical to innovation and competition in the software industry.

A. Programmers Invest Significant Time and Resources Learning the Conventions of a Particular Programming Environment.

The four computer scientists who filed an *amicus* brief in support of Oracle suggest that learning how to function in a new programming environment is a trivial exercise: “It is common for application developers to learn multiple programming languages and to create programs for competing platforms. Developers can readily adapt to and learn new languages....” Spafford Br. at 23. In fact, developing proficiency in a particular programming environment requires a significant investment of time and resources by a programmer. Peter Norvig, *Teach Yourself Programming in Ten Years*, <http://norvig.com/21-days.html> (computer science scholar criticizing programming books that purport to teach programming in weeks

when it actually takes years). While an experienced programmer can learn the conventions of a new programming environment, the switching costs in terms of time and effort may make it difficult to do so, especially if the programmer is already in the work force. If a new programming platform is created, it is in the programmer's interest for programming conventions of the new platform to be as similar as possible to those of the existing environment because this makes it easier for the programmer to work in the new programming environment. The ease of migration enhances the programmer's employment opportunities, and increases the return on the investment she made in learning the conventions of the first programming environment.

Moreover, there is a well-documented shortage of skilled programmers in the United States. In 2015, the White House announced that there were 545,000 unfilled information technology jobs. The White House, *Fact Sheet: President Obama Launches New TechHire Initiative*, Mar. 9, 2015, <https://obamawhitehouse.archives.gov/the-press-office/2015/03/09/fact-sheet-president-obama-launches-new-techhire-initiative>. To fill this gap, the White House launched a program to retrain workers to learn how to code, in addition to continuing ongoing efforts to increase the amount of computer science education in K-12 schools. *Id.* In

2016, the White House stated that the number of unfilled IT jobs is expected to double to 1 million by 2020. The White House, *Fact Sheet: A Year of Action Support Computer Science for All*, Dec. 5, 2016, <https://obamawhitehouse.archives.gov/the-press-office/2016/12/05/fact-sheet-year-action-supporting-computer-science-all>.

The shortage of skilled programmers constitutes a barrier to entry in the software industry. In order to participate in the market, a new entrant must attract programmers from other firms. Attracting programmers is particularly challenging if the programmers must learn a completely new set of programming conventions to operate in the new entrant's programming environment. Even if the new firm were willing to pay the significant costs of retraining experienced programmers, many programmers would not want to invest their time in developing new skills that might not be transferable in the future. Given the volatility of the technology sector, the new entrant may not always succeed. Experienced programmers would be reluctant to join such an enterprise if they believed it might be a career dead-end, if the expertise they gained there would not be portable.

In sum, the programmer shortage barrier to entry makes the portability of programming skills essential to the viability of new entrants.

B. Oracle Seeks to Appropriate Programmers' Sunk Costs in Learning Java.

The former Sun executives who filed an *amicus* brief in support of Oracle explain that Google's objective in replicating Sun's declaring code was "to steal the legions of developers already using the Java platform." McNealy Br. at 14. They also describe Google's "tap[ping] into the large existing base of developers already familiar with the Java APIs," *id.* at 13, as "theft." *Id.* at 14. Developers are people, not property. Oracle's copyright in the declaring code does not give it ownership of Java developers. Oracle should not be able to appropriate programmers' significant sunk cost in learning Java.

In his concurring opinion in *Borland*, Judge Boudin addressed the analogous situation of Lotus users' investment in learning how to use the Lotus command structure. He observed that a computer program's command structure "may be a creative work, but over time its importance may come to reside more in the investment that has been made by *users* in learning the menu and in building their own mini-programs — macros — in reliance upon the menu." 49 F.3d at 821 (emphasis in original). Judge Boudin added that "if Lotus is granted a monopoly on this pattern, users who have learned the command structure of Lotus 1-2-3 or devised their own macros are locked into Lotus, just as a typist who has learned the QWERTY keyboard

would be the captive of anyone who had a monopoly on the production of such a keyboard.” *Id.* He found “it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an investment in learning made by the users and not by Lotus.” *Id.* Similarly, it is hard to see why programmers who have learned the Java APIs should remain captives of Oracle because of an investment in learning made by the programmers and not by Oracle. Like Lotus, Oracle “has already reaped a substantial reward for being first.” *Id.* Furthermore, “good reasons exist for freeing” Google to attract Java programmers just as the First Circuit allowed Borland to attract Lotus 1-2-3 users: “to enable [them] to take advantage of a new advance....” *Id.*

It is not the proper role of copyright to lock programmers into a particular environment. To the extent that code needs to be copied to allow programmers to create works in new environments, such copying should be fair use. In his *Borland* concurrence, Judge Boudin suggested that Borland’s use is privileged because “it is not seeking to appropriate the advances made by Lotus’ menu; rather, ... Borland is merely trying to give former Lotus users an option to exploit their own prior investment in learning....” *Id.* A privileged use approach, based on the fair use doctrine, “would not automatically protect Borland if it had simply copied the Lotus menu (using

different codes), contributed nothing of its own, and resold Lotus under the Borland label.” *Id.* Like Borland, Google made a significant independent contribution: it constructed a new operating system optimized to function in the smartphone environment. It included the declaring code of 37 of the 166 Java API packages to allow programmers to exploit their own prior investment in learning Java.

C. Oracle’s *Amici* Misstate the Implications of this Case.

Oracle’s *amici* from the entertainment industry suggest that affirmance would encourage the unauthorized adaptation of their works to new formats, *e.g.*, converting an analog book into an e-book. *See* Copyright Alliance Br. at 14, RIAA Br. at 5, 20-21. This completely misstates the implications of this case. Google is not arguing that it was fair use for it to adapt the Java APIs’ millions of lines of code for use in the smartphone market. Rather, it is arguing that it was fair use for it to copy certain elements of the Java APIs (0.4% of the code) so that Java programmers would be able to transfer their knowledge to a new environment. What makes Google’s use fair is that it promotes additional creative activity by Java programmers by providing them a new environment in which they could apply their skills. This is precisely the sort of use the Ninth Circuit applauded in *Sega* because it would result in the “growth in creative

expression, based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote.” *Sega*, 977 F.2d at 1523.

II. Controlling Ninth Circuit Fair Use Precedent Supports a Fair Use Finding.

A. Because the Second Fair Use Factor Requires Consideration of the Nature of the Work, Ninth Circuit Fair Use Decisions Concerning Computer Programs Are the Most Relevant Precedents.

Oracle and its *amici* rely on fair use cases dealing with a wide range of copyrightable subject matter, including songs, *Campbell v. Acuff Rose Music Inc.*, 510 U.S. 569 (1994), biographies, *Harper & Row Publ’rs v. Nation Enters.*, 471 U.S. 539 (1985), television broadcasts, *Sony Corp. of Am. v. Universal City Studios*, 464 U.S. 417 (1984); *L.A. News Serv. v. KCAL-TV Channel 9*, 108 F.3d 1119 (9th Cir. 1997), unpublished letters, *Wright v. Warner Books, Inc.*, 953 F.2d 731 (2d Cir. 1991), and photographs, *Monge v. Maya Magazines, Inc.*, 688 F.3d 1164 (9th Cir. 2012).

Oracle and most of its *amici*, however, completely ignore the two most relevant Ninth Circuit precedents involving software: *Sega* and *Connectix*. The second fair use factor concerns “the nature of the copyrighted work.” Thus, right out of the gate, it is clear that fair use

decisions *involving software* would be most instructive to subsequent courts.³ As this Court recognized in *Oracle*, “some works are closer to the core of intended copyright protection than others.” *Oracle America, Inc., v. Google Inc.*, 750 F.3d 1339, 1375 (Fed. Cir. 2014) (“*Oracle II*”) (quoting *Campbell*, 510 U.S. at 586). The Ninth Circuit in *Sega* observed that “works of fiction receive greater protection than works that have strong factual elements, such as historical or biographical works, or works that have strong functional elements....” *Sega*, 977 F.2d at 1524. The *Sega* court further noted that “computer programs are, in essence, utilitarian articles—articles that accomplish tasks.” *Id.* This means that when applying fair use to a computer program, a court must be mindful of the program’s highly functional nature, and the possibility that broad copyright protection could lead to *de facto* protection for program elements outside the scope of copyright protection and other anticompetitive impacts such as lock-in. As the Ninth Circuit stated, “an attempt to monopolize the market by making it impossible for others to compete runs counter to the statutory purpose of

³ Robert Kasunic, *Is That All There Is? Reflections on the Nature of the Second Fair Use Factor*, 31 COLUM. J.L. & ARTS 529, 568 (2008) (information resulting from the second fair use factor analysis “could help formulate nuanced patterns in fair use decisions that could serve as more instructive precedent into the future”).

promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine.” *Id.* at 1523-24.

Oracle and its *amici* argue that because the infringing declaring code performs the same function in Android as it did in Java, the use is not transformative under the first fair use factor. This argument fails for two reasons. First, the Ninth Circuit found that “Connectix’s Virtual Game Station is modestly transformative,” *Connectix*, 203 F.3d at 606, even though the Virtual Game Station had similar uses and functions as the Sony PlayStation. This is because “the product creates a new platform...on which consumers can play games designed for a Sony PlayStation.” *Id.* Precisely “because the Virtual Game Station is transformative, and does not merely supplant the PlayStation console, the Virtual Game Station is a legitimate competitor in the market for platforms on which Sony and Sony-licensed games can be played.” *Id.* at 607. Accordingly, “some economic loss by Sony as a result of this competition does not compel a finding of no fair use.” *Id.* To the contrary, the Ninth Circuit found that the fourth factor favored Connectix.

Android is at least as transformative as the Virtual Game Station, and represents far more creative effort by Google than Connectix invested in the development of the Virtual Game Station. Additionally, the Virtual Game

Station provided similar uses and functions as the Sony PlayStation, while Android provides many different functions from Java.

The simplistic analysis of “transformativeness” advocated by Oracle and its *amici* also fails because under it, no use of a functional work such as a computer program could ever be transformative. The copied elements invariably would perform the same function in the new work; these elements were designed to perform a specific function. The categorically nontransformative nature of the use of functional works, in turn, would make a fair use finding *less* likely for a functional work than for a fictional work, contrary to Supreme Court precedent.

Sega and *Connectix* teach that a far more nuanced analysis of the interplay between the fair use factors is required when dealing with computer programs. The *Sega* court recognizes that its fair use conclusion “may seem incongruous at first blush.” *Sega*, 977 F.2d at 1527. After all, “Accolade, a commercial competitor of Sega’s, engaged in wholesale copying of Sega’s copyrighted code as a preliminary step in the development of a competing product.” *Id.* The *Sega* court explains that “the key to this case is that we are dealing with computer software,” and “we must avoid the temptation of trying to force ‘the proverbial square peg into a round hole.’” *Id.* (quoting *Computer Assocs. Int’l v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir.

1992)).⁴ Additionally, the Ninth Circuit stresses that “in determining whether a challenged use of the copyrighted material is fair, a court must keep in mind the public policy underlying the Copyright Act.” *Id.*

The fair use analysis in both *Sega* and *Connectix* reflects the Ninth Circuit’s awareness that rigid application of the copyright law could result in consumer “lock-in” and a decrease in the development of innovative products. The *Sega* court understands that allowing Accolade to enter the Sega-controlled Genesis market might diminish Sega’s profits and incentive to develop new products, but concluded that preventing new entrants from introducing new products was not the proper role of the copyright law. *Id.* at 1523-24. The *Sega* court recognizes that fair use provides it with a tool to address this problem: an “equitable rule of reason,” *id.* at 1522 (quoting *Harper & Row*, 471 U.S. at 560), that “permits courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which the law is designed to foster.” *Stewart v. Abend*, 495 U.S. 207, 236 (1990) (citation omitted).

Here, the jury could reasonably have concluded that copyright law should not prevent Google from creating a new platform that could readily

⁴ The *Sega* court quotes the *Altai* court’s square peg, round hole metaphor twice, *see also Sega*, 977 F.2d at 1524, underscoring the importance the Ninth Circuit attaches to the second factor, the nature of the work, in fair use cases involving software.

draw upon the existing pool of skilled Java programmers. The Java declaring code replicated in Android is part of the toolkit these programmers employ to create new apps. The millions of innovative Android apps written by these programmers provide immense benefit to consumers.⁵ The fact that Google might have been able to obtain a license to use the declaring code does not change this result; Accolade also could have obtained a license from Sega, yet the Ninth Circuit permitted Accolade's repeated reproduction and translation of the Sega Genesis operating system.

A fair use finding in favor of Google is also consistent with this Court's decision in *Chamberlain v. Skylink*, 381 F.3d 1178 (Fed. Cir. 2004). The *Skylink* court interpreted the prohibition on circumvention in 17 U.S.C. § 1201 as requiring a nexus between access and infringement for liability to attach. The court found that under a contrary rule, a manufacturer would be able "to restrict consumers' rights to use its products in conjunction with competing products." *Skylink*, 381 F.3d at 1201. This Court noted that while the antitrust laws do not negate an intellectual property owner's right to exclude others, neither do intellectual property rights confer a privilege to violate the antitrust laws. "We must harmonize" the intellectual property

⁵ As of March 2017, 2.8 million Android apps are available through Google Play. *Number of apps available in leading apps stores as of March 2017*, Statista, <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>.

laws and the antitrust law “as best we can.” *Id.* A fair use finding here allows such harmonization.⁶

In sum, this case is far more similar to *Sega* and *Connectix* than to the many fair use decisions relied upon by Oracle and its *amici* that concern artistic works where there is no possibility of lock-in or of copyright acting as a barrier to entry to legitimate competition. It is also more similar to *Sega* and *Connectix* than to the handful of fair use cases Oracle cites involving software, in which the defendants pirated entire programs and created

⁶ A fair use finding here is consistent with the Copyright Office’s discussion of fair use in its recent report on software-enabled consumer products. U.S. Copyright Office, *Software-Enabled Computer Products* (Dec. 2016). The report stated that “in many cases, copying of appropriately limited amounts of code from one software-enabled product into a competitive one for purposes of compatibility and interoperability should...be found to be a fair use.” *Id.* at 57. The report noted “even literal copying of code may be favored, if the purpose is simply to permit functionality of a software-enabled device, and not to exploit the creativity of the original author.” *Id.* at 58 (citation omitted). The Office stressed that in the section 1201 rulemaking process, it has repeatedly recognized that “interoperability is a favored purpose under the first fair use factor,” *id.*, adding that “courts also favor software interoperability when considering the second fair use factor,” because “works that are functional—like software embedded in and critical to the functioning of a consumer product—are entitled to lesser protection under the Copyright Act.” *Id.* The report concluded this discussion by stating that “proper application of these principles should ensure that copyright law preserves the ability to create interoperable products and services.” *Id.*

nothing new. *See, e.g., Wall Data Inc. v. L.A. Cty. Sheriff's Dep't*, 447 F.3d 769 (9th Cir. 2006).⁷

B. Google Had a Good Faith Belief that Replication of the Declaring Code Did Not Infringe Copyright.

Oracle and its *amici* concede that in the Ninth Circuit, a party's good faith is relevant to the fair use analysis. *See* Oracle Br. at 38; PACA Br. at 9. Google had objectively reasonable grounds for believing that replication of the declaring code did not infringe copyright.

First, after full briefing by both parties, the district court concluded in its 2012 copyrightability decision that the declaring code was not protectable. If the district court—prior to reversal by this Court—believed that the declaring code was not protectable, then surely Google had objectively reasonable grounds for believing that the declaring code fell outside the scope of copyright protection.

Second, judicial decisions prior to the development of Android could reasonably be interpreted to indicate that the declaring code was not protectable because it was necessary for interoperability. The district court's

⁷ To be sure, there are factual differences between this case and *Sega* and *Connectix*. But these factual differences cut both ways in the fair use analysis. *Sega* and *Connectix* involved intermediate copying for the purpose of detecting unprotected elements. However, *Accolade* and *Connectix* repeatedly copied and made derivative works of the entirety of the *Sega* and *Sony* programs, respectively. Google, in contrast, replicated only a small percentage of the Java API code.

2012 decision and this Court's 2015 reversal discussed the issue of interoperability at length. This Court found that the Java and Android APIs were not interoperable; no app written in Java could run on the Android platform. This is because the Android APIs included the declaring code for only 37 of the 166 Java API packages. *Oracle II*, 750 F.3d at 1372.

Nonetheless, had Google sought complete interoperability with the Java APIs, it would have had to copy *more* of the Java APIs than it did; it would have had to copy the declaring code of all (or almost all) of the 166 Java API packages. And at the time Google developed Android, there certainly was authority indicating that copyright protection did not extend to program elements necessary for interoperability. The district court cited some of these authorities in its analysis of interoperability in its copyrightability decision, *Oracle I*, 872 F. Supp. 2d at 1000 (citing *Sega* and *Connectix*). Other pre-Android authorities that could be interpreted as supporting the unprotectability of interface information include *Altai*; *Atari Games Corp. v. Nintendo of America*, 975 F.2d 832 (Fed. Cir. 1992); *Borland*; *Mitel v. Iqtel*, 124 F.3d 1366 (10th Cir. 1997); *Lexmark Int'l v. Static Control Components, Inc.*, 387 F.3d 522, 534 (6th Cir. 2004); and 17 U.S.C. § 1201(f). Of course, this Court interpreted these authorities differently from district court and Google. *Oracle II*, 750 F.3d at 1368-71.

But Google’s interpretation certainly was not unreasonable. And because Google could reasonably have believed that it was entitled to copy the declaring code for the complete set of 166 Java API packages in order to achieve interoperability, surely Google could reasonably have believed that it was entitled to copy the declaring code for just 37 Java API packages.⁸

Third, many articles published by legal scholars prior to the development of Android could reasonably be interpreted to indicate that the declaring code was not protectable.⁹ Indeed, some of these scholars believe that this Court erred in reversing the district court’s compatibility decision.¹⁰

As the *Sega* court states, “computer programs pose unique problems for the application of the ‘idea/expression distinction’ that determines the extent of copyright protection.” *Sega*, 977 F.2d at 1524. The *Sega* court

⁸ As noted by the district court in its copyrightability decision, the Connectix Virtual Game Station implemented only 137 of the PlayStation BIOS’s 242 functions. *Oracle I*, 872 F. Supp. 2d at 1000-01.

⁹ See Peter Menell, *An Epitaph for Traditional Copyright Protection of Network Features of Computer Software*, 43 Antitrust Bulletin 651, 679-92 (1998), http://works.bepress.com/peter_menell/10/ (citing a proliferation of scholarly articles on copyright and interoperability).

¹⁰ Peter Menell, *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software* (UC Berkeley Pub. Law Research Paper No. 2893192, 2017); Pamela Samuelson, *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, Berkeley Tech. L.J. (forthcoming), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2909152. See also Jonathan Band, *Interfaces on Trial 3.0: Oracle America v. Google and Beyond* (2016), <http://www.policybandwidth.com/interfaces-2-0>.

follows this statement with a lengthy discussion of the “hybrid nature” of computer programs. The inherent difficulty of separating protectable from unprotectable elements in computer programs, combined with the decisions and legal scholarship concerning interoperability, undercut the notion that Google acted in bad faith when it treated the Java declaring code as outside the scope of copyright protection.

III. The District Court Excluded Evidence That Supported Google’s Fair Use Argument.

Oracle and its *amici* complain that the district court improperly excluded evidence about the harm Android may have caused to markets other than the mobile market. Oracle Br. at 55; BSA Br. at 16. However, the district court also excluded evidence that would have benefited Google’s fair use claim. Omnibus Order on Motions in Limine for Pretrial Conference, Jan. 4, 2012.¹¹ In particular, in response to a motion in limine by Oracle, the district court excluded “any reference to the American Committee for Interoperable Systems.” Evidence concerning the American Committee for Interoperable Systems (“ACIS”) would have shown the jury that Sun and Oracle had previously taken legal positions opposite to those they were taking in this case.

¹¹ This ruling applied in the fair use trial. Joint Submission at 1, *Oracle America, Inc. v. Google Inc.*, No. 10-cv-03561 (N.D. Cal. May 9, 2016).

In the late 1980s and early 1990s, the dominant firms in the computer industry advocated interpretations of copyright law that would have restricted interoperability. Troubled by the competitive implications of these positions, leaders from new entrants convened at Sun's corporate headquarters on December 5, 1991, and founded ACIS as a response to this threat. Chaired by Sun's Deputy General Counsel Peter Choy, ACIS agreed upon a Statement of Principles, chiefly, 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", and 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...." ACIS, *Statement of Principles* (1991), available at <http://www.ccianet.org/interop>.

ACIS participated as an *amicus* in support of interoperability in many of the leading software copyright cases.¹² These included cases on non-protectability of interface specifications¹³ and the permissibility of software

¹² See CCIA, *Interoperability Resources*, <http://www.ccianet.org/interop>.

¹³ See *Altai*; *Gates v. Bando*, 9 F.3d 823 (10th Cir. 1993); *Unix Systems v. Berkeley Software*, 832 F. Supp. 790 (D.N.J. 1993); *Apple Computer v. Microsoft Corp.*, 35 F.3d 1435 (9th Cir. 1994); *Engineering Dynamics v. Structural Sys.*, 26 F.3d 1335 (5th Cir. 1994); and *Borland*.

reverse engineering.¹⁴ ACIS also lobbied for the interoperability exception to the prohibition on circumvention of technological protection measures established in 17 U.S.C. § 1201(f).¹⁵ Additionally, ACIS lobbied Congress and the Administration against other proposals that may have threatened interoperability, including legislation regarding criminal penalties for infringement of software, industrial design protection, database protection, and software patents. Further, ACIS sought a reverse engineering exception to the proposed Article 2B of the Uniform Commercial Code. *See* Jonathan Band & Masanobu Katoh, *Interfaces on Trial 2.0*, at 68 (2011).

ACIS's advocacy extended beyond the borders of the United States. ACIS argued for pro-interoperability positions in connection to international agreements such as TRIPS and the World Intellectual Property Organization

¹⁴ *See Sega; Bateman v. Mnemonics, Inc.*, 79 F.3d 1532 (11th Cir. 1996); *ProCD v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996); *Pulse Commc'ns v. DSC Commc'ns Corp.*, 528 U.S. 923 (1999); *Connectix*; and *DVD Copy Control Assoc. v. Brunner*, 113 Cal. Rptr. 2d 388 (Cal. Ct. App. 2001).

¹⁵ While the legislation was pending in Congress, Michael Morris, then Vice President and General Counsel of Sun Microsystems, argued that the exception was necessary because otherwise the legislation would impose a new "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).

Copyright Treaty, and in several Asian countries. *See Id.* at 168-75, 178-80.¹⁶

The technical issues in this case are exceedingly complex, and no doubt were challenging for a jury whose members had little programming experience. Nonetheless, the jury would have had no difficulty whatsoever understanding that the company that developed the Java APIs—Sun—had led a worldwide campaign to ensure that program elements necessary for interoperability, such as the declaring code at issue in this case, were outside the scope of copyright protection. The jury could also understand that Oracle, owner of Sun’s copyright in the Java APIs, had participated in this worldwide campaign in support of policy outcomes that now proved inconvenient. From these facts, the jury could have inferred that the actual creator of the Java APIs would never have brought this lawsuit, and that the subsequent purchaser of the copyright changed its position when it became financially expedient to do so. Oracle had articulated one interpretation of copyright law when it sought to interoperate with the systems developed by dominant companies—but it articulated the opposite interpretation once it acquired the copyright in a dominant platform. In the equitable balancing

¹⁶ The global nature of the policy battle between the entrenched dominant companies and the new entrants led to the formation of organizations parallel to ACIS in Europe, Canada, and Australia. Sun and Oracle were also members of these organizations.

underlying the fair use doctrine, the jury could have weighed this flip-flopping heavily against Oracle.

The district court judge excluded this evidence in an effort to control his courtroom and keep a technically complex case manageable for a jury of ordinary citizens. If Oracle is granted a new trial to allow evidence of the impact of Android on the desktop market, however, then Google should be allowed to introduce evidence on Oracle's reversal of its legal position.

CONCLUSION

For the forgoing reasons, the decision below should be affirmed.

Respectfully submitted,

/s/ Jonathan Band

Matthew Schruers
Vice President, Law & Policy
Computer & Communications
Industry Association
655 15th Street NW, Suite 410
Washington, D.C. 20005
(202) 783-0070
mschruers@ccianet.org

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 Amicus Curiae

May 26, 2017

CERTIFICATE OF COMPLIANCE

1. This brief complies with the type-volume limitations of Fed. R. App. P. 32(a)(7)(B) because it contains 6,431 words, excluding the parts of the brief exempted by Fed. R. App. P. 32(a)(7)(B)(iii) and Federal Circuit Rule 32(b).

2. This brief complies with the typeface requirements of Fed. R. App. P. 32(a)(5) and the types style requirements of Fed. R. App. P. 32(a)(6) because it has been prepared in a proportionally spaced typeface using Microsoft Word in 14 point Times New Roman.

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

May 26, 2017

CERTIFICATE OF SERVICE

I hereby certify, that on this 26th day of May 2017, a true and correct copy of the foregoing Brief of *Amicus Curiae* the Computer & Communications Industry Association was timely filed electronically with the Clerk of the Court using CM/ECF, which will send notification to all counsel registered to receive electronic notices.

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