Nos. 13-1021, 13-1022

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 Case No. 10-cv-03561, Judge William H. Alsup

BRIEF FOR *AMICI CURIAE* MICROSOFT CORPORATION, EMC CORPORATION, AND NETAPP, INC. IN SUPPORT OF APPELLANT

Paul T. Dacier
Krishnendu Gupta
EMC CORPORATION
176 South Street
Hopkinton, MA 01748
(508) 435-1000
Counsel for EMC Corporation

Douglas Luftman
NETAPP, INC.
495 East Java Drive
Sunnyvale, CA 94089
(408) 822-6000
Counsel for NetApp, Inc.

Gregory G. Garre
Principal Attorney
Lori Alvino McGill
LATHAM & WATKINS, LLP
555 11th Street, NW
Suite 1000
Washington, DC 20004
(202) 637-2207
Counsel for Microsoft Corporation

CERTIFICATE OF INTEREST

Pursuant to Rule 26.1 of the Federal Rules of Appellate Procedure and Rule 47.4 of the Federal Circuit Rules, counsel for amici curiae states the following:

Amicus Microsoft Corporation ("Microsoft") is a publicly held corporation. Microsoft does not have a parent corporation and no publicly held corporation holds 10% or more of its stock. Microsoft did not participate in this case in the district court.

Amicus EMC Corporation ("EMC") is a publicly held corporation. EMC does not have a parent corporation and no publicly held corporation holds 10% or more of its stock. EMC did not participate in this case in the district court.

Amicus NetApp, Inc. ("NetApp") is a publicly held corporation. NetApp does not have a parent corporation and no publicly held corporation holds 10% or more of its stock. NetApp did not participate in this case in the district court.

The following are the names of all the law firms and attorneys expected to appear for Microsoft, EMC, and NetApp in this Court:

Gregory G. Garre, LATHAM & WATKINS LLP.

Lori Alvino McGill, LATHAM & WATKINS LLP.

Paul T. Dacier, EMC CORPORATION.

Krishnendu Gupta, EMC CORPORATION.

Douglas Luftman, NETAPP, INC.

/s/ Gregory G. Garre
Gregory G. Garre
Principal Attorney
Lori Alvino McGill
LATHAM & WATKINS, LLP
555 11th Street, NW
Suite 1000
Washington, DC 20004
(202) 637-2207
Counsel for Microsoft Corporation

Paul T. Dacier
Krishnendu Gupta
EMC CORPORATION
176 South Street
Hopkinton, MA 01748
(508) 435-1000
Counsel for EMC Corporation

Douglas Luftman
NETAPP, INC.
495 East Java Drive
Sunnyvale, CA 94089
(408) 822-6000
Counsel for NetApp, Inc.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES AND RULES

ADDITIONAL AUTHORITIES

STATEMENT OF INTEREST[1]

This case tests the copyrightability of computer programs, specifically packages of source code that are part of the Java software platform used by third-party software developers to write applications for computers, tablets, smartphones, and other devices running Java. *Amicus* Microsoft Corporation ("Microsoft") is a leading innovator in computer software, and has spent nearly forty years creating software platforms for application developers, including the well-known Windows operating system. *Amicus* EMC Corporation ("EMC") is the world's largest provider of storage systems, software, and solutions that help customers store, manage, protect and analyze information and data in a more agile, trusted and cost-efficient way. *Amicus* NetApp, Inc. ("NetApp") is a NASDAQ-100, Fortune 500 technology company with a twenty-year history of innovation in operating systems, data management and other software markets.

The district court's holding that critical elements of the software platform at issue in this case are not copyrightable at all is the product of several significant errors of copyright law. In particular, the court failed to appreciate the key distinction between the threshold question of what is copyrightable (which, as the Supreme Court has made clear, is subject to an exceedingly low bar) and the role

[1] No party's counsel authored this brief in whole or in part. No party, party's counsel, or any person other than *amicus* or its counsel contributed money intended to fund preparing or submitting this brief. *See* Fed. R. App. P. 29(c)(5).

of separate doctrines to determine when a valid copyright has been infringed. If allowed to stand, the district court's ruling would upset settled expectations and harm incentives for innovation in the software industry.

Microsoft's mission is to enable individuals and businesses throughout the world to realize their full potential by creating technology that transforms the way people work, play, and communicate. Microsoft develops, manufactures, licenses, and supports a wide range of programs and services, including Windows, Microsoft Office and Microsoft Office 365, Xbox and Xbox Live, and Bing. Microsoft invests billions of dollars in research, development, and promotion of new technologies, products, and services, and competes vigorously in dynamic technology markets.

EMC is a global leader in enabling businesses and service providers to transform their operations and deliver information technology as a service. The company manufactures, develops, and sells a comprehensive, best-of-breed portfolio of data storage systems and software, security management software, and, through its majority equity stake in VMware, Inc. ("VMware"), is the leading provider of virtualization and virtualization-based cloud infrastructure software solutions. EMC employs approximately 60,000 people around the globe, including thousands of software developers, invests billions of dollars in research and development annually in information technology solutions, and earns billions of dollars annually in software revenue from products related to the company’s information infrastructure and VMware virtual infrastructure portfolio. EMC’s leadership and investment in innovation are exemplified by the company's ranking as one of the ten largest software companies in the world, its standing as one of Thomson Reuters’ Top 100 Global Innovators in 2012, and its top ten ranking on the Patent Board’s Information Technology industry scorecard.

NetApp is a leading vendor of innovative storage and data management solutions that form the foundation for efficient and flexible information technology infrastructures. NetApp’s products include, among many others, storage systems—such as its fabric-attached storage (FAS) and E-Series product lines that help customers streamline operations and lower the cost associated with storing and managing their data—and operating system software—such as the Clustered Data ONTAP® system, based on NetApp’s patented WAFL® technology. NetApp’s tradition of innovation, particularly in the software industry, is further reflected in not only its robust intellectual property portfolio, but also by its recognition as one of the World’s Most Innovative Companies (Forbes), a Top 300 Patent Holder (Intellectual Property Owners Association), and the owner of one of the top quality portfolios in its industry (IEEE Spectrum).

Amici are all keenly interested in both preserving settled copyright law and sensibly resolving new issues, in order for the domestic intellectual property regime to function fairly, efficiently, and predictably. And *amici* are well-suited to address the broader legal, economic, technological, and social implications of the important question presented by this case in particular. On the one hand, *amici* have relied on copyright protection to help develop and license some of the most successful software products in history. On the other hand, *amici* are among the world's largest *users and licensees* of copyrighted works, including software, and have a longstanding strategic interest in preserving room for legitimate reverse-engineering, competitive analysis, and innovative follow-on development of existing software. *Amici* have customers with the critical need for their products to interoperate effectively with products provided by other vendors, including those provided by other *amici.* Toward this end, *amici* must be able to carefully and securely control deployment of their own copyrighted works, and at the same time be able to use systems, platforms, infrastructures, and solutions built from connectable offerings provided by multiple vendors. Further, *amici* actively use, contribute to, and sponsor open source projects.

Amici regularly play the role of both parties to the business transaction at the heart of this case: like Oracle America, Inc. ("Oracle"), *amici* have created and maintain numerous technological ecosystems for third-party developers to work in; and like Google, Inc. ("Google"), *amici* frequently seek to interoperate with and build on innovations created by others, whether through software licenses or by reusing facts, ideas, and other elements of existing works that are not protected by copyright law. *Amici* frequently compete and collaborate with each other and other vendors, in relationships defined in part by established copyright law.

Although *amici* do not take a position on the ultimate question of whether the software packages at issue in this case are copyrightable and whether any copyright has been infringed, *amici* urge this Court (1) to hold that the district court's copyright analysis was fundamentally flawed and (2) to decide this case in light of the settled copyright principles discussed below.[2]

INTRODUCTION AND SUMMARY OF ARGUMENT

Congress has determined that computer software is eligible for copyright protection. 17 U.S.C. § 101. Copyright protects computer software in several important respects. It covers the literal lines of code that comprise software, generally preventing their reproduction or distribution without permission from the rightsholder. But copyright also covers certain *non-literal* elements of the software as well. For example, the "structure, sequence, and organization" of a software product—above and beyond the 1s and 0s that make up the program at its literal level or the exact words of the human-readable source code—can, in some instances, be protected by the copyright in the work. As a result, copyright infringement in a software case can occur even when the defendant did not copy

[2] The parties have consented to the filing of this brief.

the underlying developers' code, where the defendant has copied some other, non-literal element of the software subject to copyright protection.

There is a critical difference, however, between the ultimate question of whether copyright in a software product has been *infringed*, and the threshold question of what elements of the plaintiff's software may be *copyrighted* in the first place. These two questions implicate substantially different principles of copyright law and considerations of innovation policy. "The *sine qua non* of copyright is originality," and only a "modicum of creativity" is required for a work to be sufficiently original to qualify for copyright protection. *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345, 346 (1991). As the Supreme Court stressed in its seminal decision in *Feist,* "the originality requirement is not particularly stringent." *Id.* at 358. Under settled doctrine, a court must look not just to whether individual elements of a work merit protection when viewed in isolation, but also to whether they reflect copyrightable creativity when viewed in the aggregate, as an original *collection* of independent elements—even if each element would be unprotectable on its own.

At the same time, just because a work (or combination of elements) is copyrightable does not mean that copying it will result in a finding of *infringement*. For example, concepts like the fair use doctrine, the requirement that works be "substantially similar" in order to be infringing, and the merger doctrine all sanction an appropriately broad and flexible range of competing and complementary innovation based on or inspired by any existing software product. Copyright law thus balances the need to incentivize a first-mover's independent creation with the need to allow follow-on users to innovate in their own right.

The district court below fundamentally erred in reaching its conclusion that the Java software packages Google admittedly copied are not copyrightable at all. The court failed to distinguish between cases addressing *infringement* and cases addressing *copyrightability* in the first instance. It failed to properly apply the low originality standard the Supreme Court has prescribed. And it failed even to consider whether any collection of elements in the Java software platform—including the names of its methods and classes, the organization and hierarchy of its packages and their constitutive parts, and the selection of features in and across packages—represents a copyrightably original *combination*.[3] Instead, the court applied an atomistic test of its own creation that considered the copyrightability of each discrete component of Oracle's software packages separately, as if it were

[3] As Oracle's opening brief explains (at 21), the terminology used in the case is confusing. The Java software packages at issue here are called "Application Programming Interfaces," or "APIs." The term API is used in the software industry to describe a wide range of things, some of which are very simple and some of which are very complex, and each of which has different purposes and context. To provide clarity, *amici* refer to the computer programs here as "software packages" or "platforms." *Amici* do not address APIs beyond the computer programs at issue here.

analyzing the copyrightability of a book by considering each word alone, rather than in concert with the surrounding ones.

That decision sets a dangerous and ill-advised precedent. Under established precedent, sufficiently original software packages like those in the Java platform certainly *may* be copyrightable, preventing free-riders from replicating their precise structure and suite of features. Yet the district court's reasoning leaves no room for that result—not only in this case but on virtually *any* facts. To be clear, *amici* do not suggest that those elements of *every* computer program are copyrightable, or that copyright in Oracle's Java platform would prevent second-comers from using the platform to foster further software development or create competing products. Even for copyrightable platforms and software packages, the determination whether infringement has occurred must take into account doctrines—like fair use—that protect the legitimate interests of follow-on users to innovate. But the promise of *some* threshold copyright protection for platforms like Java specifically and other elements of computer software generally is a critically important driver of research and investment by companies like *amici*—and rescinding that promise would have sweeping and harmful effects throughout the software industry.

This brief focuses on the legal errors the district court committed in analyzing copyrightability and the damaging implications for innovation policy that would result from its copyrightability ruling. *Amici* do not address the ultimate question of whether the software packages at issue are copyrightable under the proper legal analysis, whether this Court should reverse outright or remand for reconsideration in light of the proper legal analysis, or the merits of Google's "fair use" defense or other issues concerning infringement *vel non*.

ARGUMENT

I. THE DISTRICT COURT'S HOLDING THAT THE JAVA PLATFORM WAS NOT COPYRIGHTABLE AT ALL WAS FLAWED AS A MATTER OF COPYRIGHT LAW AND POLICY, AND WOULD DESTABILIZE THE SOFTWARE INDUSTRY

Copyright protects all the elements of a work that satisfy the Supreme Court's "modicum of creativity" standard. *See Feist*, 499 U.S. at 362; *Boisson v. Banian, Ltd.*, 273 F.3d 262, 268 (2d Cir. 2001); *Urantia Found. v. Maaherra*, 114 F.3d 955, 959 (9th Cir. 1997). The originality standard is "extremely low"—"even a slight amount [of creativity] will suffice." *Feist*, 499 U.S. at 345. This bedrock principle applies no less to the copyright in a work of computer software than any other type of work. *See, e.g.*, *Softel, Inc. v. Dragon Med. & Scientific Commc'ns, Inc.*, 118 F.3d 955, 964 (2d Cir. 1997) (applying the *Feist* copyrightability analysis to computer software); *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1445 (9th Cir. 1994) (same). Computer software can be copyrightably original even if some of its constitutive building blocks are not copyrightable in their own right. Indeed, "[i]n *Feist*, the Court made quite clear that a compilation of non-protect[a]ble elements can enjoy copyright protection even though its constituent elements do not." *Softel*, 118 F.3d at 964.

A. The District Court Misapplied And Disregarded Basic Copyright Principles

The district court in this case lost sight of these principles. First, it analyzed copyrightability not by looking to the plaintiff's work as a whole, but instead by considering *only* the portion of the plaintiff's work that Google admittedly copied. *See Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 997-98 (N.D. Cal. 2012) (addressing only the 37 software packages that Google copied, not the 166 that Java contains). This mode of analysis is backwards. Under both prevailing doctrine and common sense, assessing copyrightability requires examining the allegedly infringed work *in full*. *See Softel*, 118 F.3d at 964; *Boisson*, 273 F.3d at 269-71; *Apple Computer*, 35 F.3d at 1442. Once a court has determined that a plaintiff's copyright is valid and relevant, it can and should turn its attention to the defendant's copycat work to decide whether any *infringement* has occurred. *See Apple Computer*, 35 F.3d at 1442. But by examining only the part of the allegedly infringed work that tracks part of the allegedly infringing work, the district court failed to consider whether and to what extent the allegedly infringed work, *in total*, merits copyright protection. The court thus proceeded from the wrong baseline from the very start in deciding copyrightability.

Second, the district court failed entirely to consider whether the software packages in the Java platform are copyrightable as an original collection and organization of discrete elements, irrespective of whether each discrete element is copyrightable in its own right. *See Softel*, 118 F.3d at 964. That by itself was reversible error. As the Second Circuit has taken pains to explain, "'[a]n original arrangement of uncopyrightable or public domain works—even facts—is as copyrightable as a compilation in the computer context as it is elsewhere in copyright law.'" *Id.* (quoting Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977, 1003 (1993)); *see also Apple Computer*, 35 F.3d at 1445 ("[O]riginal selection and arrangement of otherwise uncopyrightable components may be protectable.").

The district court neglected to consider the relationships among the 166 packages in the Java platform and the methods they contain, and the particular selection of features and functions they include, in the particular hierarchy in which they are embedded. Instead, the district court excised discrete elements of the platform and assessed them individually and in isolation. It concluded that because names and short phrases generally are uncopyrightable independently, all of the names in Java's thousands of methods (no matter how they were identified, selected, arranged, and deployed by Oracle) were unprotected. 872 F. Supp. 2d at

999. It concluded that "[f]unctional elements essential for interoperability are not copyrightable," and erroneously extended that conclusion to find that not a single such element could contribute to the copyrightability of the Java platform as a whole. *Id.* at 997. And it (apparently) concluded that elements of the platform could not be protected by copyright under the "merger" doctrine, so excluded them from the copyrightability analysis. *Id.*

The impermissibly stringent originality standard imposed by the district court is reflected in its discussion of *American Dental Ass'n v. Delta Dental Plans Ass'n*, 126 F.3d 977 (7th Cir. 1997). *See Oracle*, 872 F. Supp. 2d at 998-1001. *Delta Dental* concerned a *taxonomy* categorizing a *pre-existing* set of dental procedures, not the copyrightability of computer software or of any originally selected set of previously unconnected elements in general. As *Delta Dental* itself explains, a "taxonomy" (in the copyright law sense) reflects *no* creativity in the selection of elements it includes; by definition, a taxonomy is simply an attempt to comprehensively categorize a field. *See* 126 F.3d at 980. The only potentially copyrightable originality in a taxonomy is in the organization and arrangement of its pre-defined constitutive parts. Generally speaking, taxonomies therefore present a much harder case for establishing copyrightability than does computer software.[4] The Java platform contains a category of creative originality not present in a taxonomy: the choice of *what* to include in it. The district court's lengthy attempt to distinguish the reasoning in *Delta Dental* thus reveals the flawed premises underlying its copyrightability analysis.

The district court also surveyed cases such as *Sega Enterprises, Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) and *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992). *See Oracle*, 872 F. Supp. 2d at 993-94. These are certainly canonical software copyright opinions—but they do not concern the question whether works like the Java packages are effectively unprotected by copyright altogether. Instead, they deal with software works that are plainly protected, and address whether a second-comer *infringed* the rightsholder's copyright interest by borrowing more than was permissible for a follow-on use. *See Sega*, 977 F.2d at 1514 (considering whether reverse-

[4] Nevertheless, even taxonomies may be copyrightable. And, indeed, *Delta Dental* found the taxonomy at issue in that case sufficiently original to be copyrightable—though other courts to consider the copyrightability of taxonomies have reached different results. *Compare Delta Dental*, 126 F.3d at 979 *with Southco, Inc. v. Kanebridge Corp.*, 390 F.3d 276 (3d Cir. 2004) (en banc) (numbering system used to categorize manufacturer's product suite of screws and fasteners not copyrightable) *and ATC Distribution Group, Inc. v. Whatever It Takes Transmissions & Parts, Inc.*, 402 F.3d 700 (6th Cir. 2005) (system of organization in automobile transmission parts catalogue not copyrightable).

engineering copyrighted software amounted to "fair use"); *Atari*, 975 F.2d at 844 (same).

Cases that the district court ignored are more instructive for the threshold question of copyrightability raised here. For example, the district court overlooked *Softel* and *Apple Computer*, both of which extensively discuss the copyrightability of critical elements of computer software, and even *Boisson* and *Tufenkian Import/Export Ventures, Inc. v. Einstein Moomjy, Inc.*, 338 F.3d 127 (2d Cir. 2003), which concern copyright protection for compilations of unprotected elements in the context of tangible goods such as quilts (*Boisson*) and rugs (*Tufenkian*). Generally speaking, complex, highly developed computer software is plainly copyrightable under the principles these cases apply.

In creating the Java platform, Oracle did not simply categorize some pre-existing set of method names, software features, or API packages.[5] Instead, it compiled its own chosen collection of names, and features, and packages into a work of software, with a particular organization and set of hierarchies, of its own selection and design. *See* Appellant's Br. at 24-25. Even if each individual element of that work is not protected by copyright in isolation, the copyrightability

[5] Sun Microsystems of course was the party that actually created the Java API. We use the term "Oracle" here for ease of reference, given that Oracle Corporation acquired Sun and renamed it Oracle America, Inc.

analysis mandated by long-settled precedent requires an additional examination of the originality of the work as an aggregation of all of its elements. The district court failed to conduct that analysis.

B. The District Court's Copyrightability Ruling Would Upset Settled Expectations And Harm Incentives For Innovation

Affirming the district court's ruling on copyrightability would needlessly undermine the foundation on which extensive and widely beneficial licensing ecosystems have been built throughout the software industry. Creators of software platforms—from proprietary operating systems like Microsoft Windows, NetApp's Clustered Data ONTAP, and EMC's Enginuity and OneFS to open source platforms like Linux—*all* depend on copyright as the lynch-pin of their operation. "Like other information goods, software platforms are … reproducible at virtually no cost." David S. Evans, Andrei Hagiu, and Richard Schmalensee, *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries* 79 (2006). Yet they "create value by reducing the costs for their multiple customer groups to come together and thereby enhance the value each customer group delivers to the other. They do this mainly through providing shared services—made available through APIs—that reduce costs for developers and users." *Id.*

Copyright licensing thus provides the solution for a dilemma faced by software platform creators: how to provide those "shared services … through

APIs" to users without enabling threats to the purposes and goals of that platform. The lessons from the open source experience are instructive:

> The proponents of open-source software faced a problem. On the one hand they wanted to make open-source software widely available. That meant that they did not want to use copyrights, patents, or trade secrets to limit the distribution of open-source programs. On the other hand, they wanted to make sure that commercial enterprises could not free-ride on the efforts of the open-source community by making minor changes or additions to open-source programs but then enforcing their own intellectual property rights on the entire modified programs. The General Public License (GPL) was an ingenious solution to this dilemma. … Despite the copyleft name, the GPL is enforced by copyright law. *Copyright is the source of the property protection that enables those who release software under a GPL to impose conditions on others who obtain that code.*

Id. at 74 (emphasis added).

The chief legal reason why users must abide by the terms of the General Public License, or the Mozilla Public License, or *any* open-source license, is that failing to do so exposes the violator to potential copyright liability. *See Jacobsen v. Katzer*, 535 F.3d 1373 (Fed. Cir. 2008); Natasha Horne, Note & Comment, *Open Source Software Licensing: Using Copyright Law to Encourage Free Use*, 17 Ga. St. U. L. Rev. 863 (2001); Greg R. Vetter, *The Collaborative Integrity of Open-Source Software*, 2004 Utah L. Rev. 563, 574. By the same token, the reason why companies like *amici* are able to maintain open platforms for third-party developers is that they can count on copyright as the primary mechanism for enforcing a consistent set of practices that maximize productivity, interoperability, and distributed innovation.

Without copyright, *amici* and similarly situated companies would be forced to retrench and only disclose information about their computer programs to users and developers in more restrictive and costly ways, to prevent harmful misappropriation of their innovation or threats to the integrity of the platform. That would strike a devastating blow to innovation in the software industry, cross-industry collaboration and ultimately to downstream consumers.

The district court purported to limit its copyrightability ruling to the "specific facts of this case." *Oracle*, 872 F. Supp. 2d at 1002. But if the software packages in the Java platform—a sophisticated, complex, and successful programming interface—are not "copyrightable in the first place," *id.* at 997, then it is difficult to conceive of what platform would qualify for copyright protection under the decision below. That is particularly true given that the district court's analysis of copyrightability essentially disregards any consideration of whether a software work is copyrightable as an original collection and organization of discrete elements, irrespective of whether each discrete is copyrightable in its own right. In any event, the message of the decision below will be loud and clear when it comes to the impact on innovation for firms or individuals investing or engaged in the development of software in this vitally important area—in which the United States has led the world in innovation.

II. COPYRIGHT LAW ACCOUNTS FOR THE FREEDOM TO INNOVATE WITH FOLLOW-ON CREATIONS THROUGH DOCTRINES THAT ARE MORE EFFECTIVE AND BALANCED THAN THE DISTRICT COURT'S ALL-OR-NOTHING APPROACH TO THE COPYRIGHTABILITY OF SOFTWARE PLATFORMS

Recognizing a copyright in the Java platform would not necessarily mean that Google's conceded copying would amount to infringement. That determination depends on a second level of analysis that considers venerable copyright doctrines such as fair use, the requirement of substantial similarity for infringement, and the principle of idea/expression "merger." Those doctrines can—and regularly do—vindicate legitimate concerns about the freedom to innovate and build competitive software products. They do so, however, in a more robust and balanced way than the district court's flawed, all-or-nothing approach to the threshold question of whether Oracle's software was copyrightable at all.

Amici are strong supporters of the settled exceptions and limitations to copyright protection. Legal principles that preserve competition and facilitate follow-on innovation by constricting the scope of copyright protection are both necessary and plentiful in U.S. copyright law. They include, among others:

- The fair use doctrine (*see* 17 U.S.C. § 107);

- The principle that bare methods of operation are unprotected *ab initio* (*see* 17 U.S.C. § 102(b));

- The merger doctrine (*see Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 706 (2d Cir. 1992)); and

- The rule that copyright protection does not extend to portions of a work that are the creation of someone other than its author, including but not limited to *scenes à faire* (*see Softel*, 118 F.3d at 964).

These are just some of the conventional limitations and exceptions in copyright law relevant to the particular circumstance of this litigation.

Courts regularly invoke these principles to find that reuses of significant portions of copyrighted works of computer software are permissible, particularly where the reuse is intended to achieve interoperability between existing computer programs and new programs and technology the defendant developed. For example, in *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596, 608 (9th Cir. 2000), the Ninth Circuit found that it was "fair use" for a follow-on innovator to reproduce the copyright-protected portions of a work of computer software in order to reverse-engineer elements of the work that fell outside the scope of copyright protection. In the same vein, in *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965, 971 (9th Cir. 1992), the Ninth Circuit held it was fair use for consumers to use a third-party, add-on product to alter the copyright-protected displays of Nintendo video games. And in *Sega*, 977 F.2d at

1526, the court of appeals concluded that fair use permitted copying plainly protected software object code specifically to create competing products. The list goes on.[6]

As these cases show, the fact that a work of computer software is copyrightable in the first instance does *not* mean that no one can make a competing product, or a product that invokes similar tools and methods, or a product that interoperates with or performs a similar set of functions as a copyrighted software work, or a product that builds on the state of the art to offer users a newer and better experience. And as vigorous competitors in numerous software markets, *amici* would not have it any other way. The district court was therefore quite wrong in saying that, "[t]o accept Oracle's claim would be to allow anyone to copyright one version of code to carry out a system of commands and thereby bar all others from writing their own or different versions to carry out all or part of the same commands." *Oracle*, 872 F. Supp. 2d at 1002.

[6] In contrast to these cases, in the present litigation, the "compatibility" that Google sought to leverage was not with Oracle's platform itself, but instead with *developers' pre-existing experience and comfort with Java*. *See* Appellant's Br. at 65-66. In other words, though each of the precedents discussed here concerns a situation in which copying plausibly redounded to the benefit of the original platform creator, there is no analogous benefit to Oracle in this case, since Google copied Java purely in the interest of making its own competing product.

The fact that a work of computer software is protected by copyright *does*, however, mean that the range of permissible follow-on uses is shaped by centuries' worth of copyright law principles. These doctrines form the legal backdrop against which software development takes place. To hold that copyright protection simply never attaches to software platforms *at all* would be to replace that deep well of nuanced (if occasionally contestable) legal rules and principles with nothing. It would be to throw away the long-evolving, largely balanced approach to technological innovation that copyright law has settled into after generations of wrangling and accounting for novel developments in new contexts. These rules are not perfect, but they are flexible and they are known, and companies and business practices have grown up in reliance on them, including in the software industry.

Amici express no view on whether Google's challenged conduct ultimately amounts to infringement, factoring in "fair use" and other limiting principles. Instead, *amici* simply stress that there exists a robust array of doctrines that allow courts to balance the competing interests of incentivizing innovation in the first instance, and permitting legitimate and necessary subsequent competition and innovation. The district court's apparent attempt to use copyrightability—alone—as an all-purpose tool to achieve those complex and subtle goals, *see* 872 F. Supp. 2d at 1001-02, was contrary both to existing law and to sound policy designed to protect and foster innovation in this vital area. The court's ruling, if allowed to stand, will deal a serious if not potentially staggering blow to the existing incentives established by copyright law for innovation in this critical industry.

CONCLUSION

For the foregoing reasons, *amici* urge this Court to hold that the district court's copyrightability analysis was erroneous, and to resolve this case in accordance with the settled legal principles discussed above.

February 19, 2013

Respectfully submitted,

/s/ Gregory G. Garre
Gregory G. Garre
Principal Attorney
Lori Alvino McGill
LATHAM & WATKINS, LLP
555 11th Street, NW
Suite 1000
Washington, DC 20004
(202) 637-2207
Counsel for Microsoft Corporation

Paul T. Dacier
Krishnendu Gupta
EMC CORPORATION
176 South Street
Hopkinton, MA 01748
(508) 435-1000
Counsel for EMC Corporation

Douglas Luftman
NETAPP, INC.
495 East Java Drive
Sunnyvale, CA 94089
(408) 822-6000
Counsel for NetApp, Inc.

CERTIFICATE OF COMPLIANCE

I hereby certify that this brief complies with the type-volume limitations of Fed. R. of App. P. 29(d), because the brief contains 4,965 words, excluding the parts of the brief exempted by Fed. R. App. P. 32(a)(7)(B)(iii).

I further certify that this brief complies with the typeface requirements of Fed. R. App. P. 29(c) and 32(a)(5), because this brief was prepared using Microsoft Word 2003 in 14-point Times New Roman font.

/s/ Gregory G. Garre
Gregory G. Garre