NO. 13-1021, -1022

In the United States Court of Appeals for the Federal Circuit

ORACLE AMERICA, INC.,

Plaintiff – Appellant,

v.

GOOGLE INC.,

Defendant – Cross-Appellant.

BRIEF OF AMICI CURIAE RACKSPACE US, INC., APPLICATION DEVELOPERS ALLIANCE, TMSOFT, LLC, AND STACK EXCHANGE INC.

Appeal from the United States District Court for the Northern District of California Case No. 10-CV-3561, Hon. William H. Alsup

Chad Ruback
The Ruback Law Firm
8117 Preston Road, Suite 300
Dallas, Texas 75225
(214) 522-4243
(214) 522-2191 fax

Attorney for Amici Curiae, Rackspace US, Inc.,
Application Developers Alliance, TMSOFT, LLC, and Stack Exchange Inc.

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rules 21(a)(2) and 47.4(a)(1), counsel for the Amici Curiae Rackspace US, Inc., Application Developers Alliance, TMSOFT, LLC, and Stack Exchange Inc. certifies the following:

1. The full name of every party represented by me:

   Rackspace US, Inc.

   Application Developers Alliance

   TMSOFT, LLC

   Stack Exchange Inc.

2. The names of the real parties in interest I represent are:

   Rackspace US, Inc.

   Application Developers Alliance

   TMSOFT, LLC

   Stack Exchange Inc.

3. All parent corporations, and any publicly held companies that own 10 percent or more of the stock of the parties I represent are:

   Rackspace Hosting, Inc. is a parent corporation of Rackspace US, Inc. No publicly held companies own 10 percent or more of Rackspace US, Inc.'s stock.

   Application Developers Alliance does not have a parent corporation, and no publicly held companies own 10 percent or more of the Alliance's stock.

   TMSOFT, LLC does not have a parent corporation, and no publicly held companies own 10 percent or more of TMSOFT, LLC's stock.

   Stack Exchange Inc. does not have a parent corporation, and no publicly held companies own 10 percent or more of Stack Exchange Inc.'s stock.

4. The names of all law firms and the partners or associates that appeared in the trial court or are expected to appear in this court for the parties I now represent are:

   Chad Ruback
   The Ruback Law Firm

May 30, 2013

/s/ Chad Ruback
Chad Ruback
Attorney for Amici Curiae
Rackspace US, Inc., Application
Developers Alliance, TMSOFT, LLC,
and Stack Exchange, Inc.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES AND RULES

OTHER AUTHORITIES

Google, *Google Buzz Has Gone Away, But Your Posts Are Yours to Keep*, https://support.google.com/mail/answer/1698228?hl=en (last visited May 29, 2013) 18

Andrew B. Hebl, Note, *A Heavy Burden: Proper Application of Copyright's Merger and* Scenes A Faire *Doctrines*, 8 Wake Forest Intell. Prop. L.J. 128 (2007) 21

Jacqueline D. Lipton, *IP's Problem Child: Shifting the Paradigms for Software Protection*, 58 Hastings L.J. 205 (2006) 20

Open Handset Alliance, *Android FAQ*, http://www.openhandsetalliance.com/android_overview.html (last visited May 29, 2013) 18

OPEN SOURCE INITIATIVE, http://opensource.org/osd (last visited May 29, 2013) 12

Openstack Cloud Software, *What is OpenStack?*, http://www.openstack.org/ (last visited May 29, 2013) 17

Press Release, Europa, Mergers: Commission Opens In-Depth Investigation Into Proposed Takeover of Sun Microsystems by Oracle (Sept. 3, 2009), http://europa.eu/rapid/press-release_IP-09-1271_en.htm?locale=en 14

Press Release, Oracle Makes Commitments to Customers, Developers and Users of MySQL (Dec. 14, 2009), http://www.oracle.com/us/corporate/press/042364 14

SERVICES FOR UNIX—INTEROPERABILITY BLOG, http://blogs.msdn.com/b/sfu/ (last visited May 29, 2013) 11

Andrew S. Tanenbaum, *Some Notes on the "Who wrote Linux" Kerfuffle, Release 1.5* (May 20, 2004) http://www.cs.vu.nl/~ast/brown/ 11

Timothy S. Teter, Note, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 Stanford L. Rev. 1061 (1993) 16, 21

Stephen R. Walli, *INTERIX: UNIX: Application Portability to Windows NT via an Alternative Environment Subsystem*, http://stephesblog.blogs.com/papers/usenix-interix.pdf 11

INTERESTS OF AMICI CURIAE[1]

Rackspace is a cloud-computing company that delivers enterprise-level hosting services to business of all sizes and kinds throughout the world. Since its founding in 1998, Rackspace has grown from a small company based in San Antonio, Texas to an industry pioneer serving more than 200,000 customers in over 120 countries. One of Rackspace's greatest achievements is its collaboration with NASA to co-found OpenStack, the world's fastest-growing open source cloud platform and developer community. OpenStack provides an alternative to the proprietary software that powers most other major cloud computing platforms by making its code freely available to developers.

Rackspace's involvement in both the open source and cloud computing markets lends the Court a unique perspective on the copyright issues raised in this appeal. Open source development is a driving force behind software innovation and fosters the creation of peer-reviewed, higher-quality code. The open source community is a perfect example of why copyright protection for APIs is unnecessary for continued innovation and why the monopoly Oracle seeks would in fact hamper that innovation. Further, the cloud computing market depends heavily on the use of APIs because all cloud functionality is delivered to the end user via APIs.

[1] The parties have consented to the filing of this brief pursuant to Fed. R. App. P. 29(a). No party's counsel authored this brief in whole or in part. No party, party's counsel, or any person other than amici or their counsel contributed money intended to fund preparing or submitting this brief. *See* Fed. R. App. P. 29(c)(5).

Accordingly, Rackspace has extensive experience with APIs and their inherently functional nature.

The Application Developers Alliance ("Alliance") is a global association of more than 25,000 individual software developers and more than 125 companies who design and build applications ("apps") for consumers to use on mobile devices like smartphones and tablets.[2] TMSOFT is one of the Alliance's corporate members and is a small app publisher of several popular apps including sleep aids, music visualization, and games. Apps run on software platforms, including Google's Android, Apple's iOS, and Facebook, and are sold or distributed through virtual stores like Google's Play Store, Apple's App Store, Amazon.com, and Handango. TMSOFT's White Noise app has been widely downloaded on all major app platforms, and was the most popular app in the Apple App Store after its debut in 2008.

The Alliance was formed to promote continued growth and innovation in the rapidly-growing app industry and routinely speaks as the industry's voice to legislators and policy-makers. Millions of Americans and tens of millions of individuals world-wide earn their livelihood through app development. App

[2] The views expressed in this brief are the Alliance's own and may or may not represent the views of each individual member.

development is one of the fastest growing industries in America and has transformed the software industry.

As described in more detail below, app development requires programmers to use API declaring code associated with individual operating system platforms. The ability to freely exchange and use the API declaring code is vital to Alliance members' and TMSOFT's ability to develop the innovative applications that serve sectors as diverse as healthcare, fashion, fitness, and public services. Accordingly, the Alliance, TMSOFT, and the Alliance's other members have wide-ranging experience with APIs and can attest to the use of APIs in the app and mobile device industry and in the programming community at large.

Stack Exchange is a network of technical communities, each dedicated to serving experts in a specific field. As part of its support for developer and creator communities, Stack Exchange builds and maintains libraries of high-quality questions and answers, focused on each community's area of expertise. It is the most-frequented developer site on the Internet, with over 100 separate sites and over 64 million monthly unique visitors.

Stack Exchange has a close relationship with APIs. From programmers sharing answers on parsing HTML, to researchers seeking solutions to combinatorial problems, Stack Exchange's communities are focused on helping each other explore, describe, and implement APIs. Just as photographers compare their favorite cameras and gardeners discuss seeds, APIs are the tools of the trade for the majority of Stack Exchange's visitors, including APIs for mobile development. As a company focused on the nuances of API construction and use, Stack Exchange has a unique perspective informed by the thousands of software development experts that participate every day on its sites.

SUMMARY OF ARGUMENT

Software is a mix of both functional and expressive components. For the past forty years, the common understanding in the industry has been that the "declaring code"—the code used to define interfaces and APIs—is functional and not copyrightable. In contrast, the "implementing code"—the code needed to provide the underlying functionality—is expressive and protected by copyright.[3] Oracle's position, that the declaring code used to define an API is protectable under copyright, is contrary to the current and historical practice and expectations in the software industry and would have a devastating impact on the software business as a whole. Thousands of businesses, developers, and even end users would be faced with legal uncertainty if declaring code were afforded copyright protections.

[3] Amici use the term "declaring code" as that term has been defined by Oracle and is distinguished from the term "implementing code." (Oracle Br. 10). As defined by the district court, the term implementing code refers to the code which carries out the function defined by the API. (A163). The district court found that "[t]o carry out any given function, the method specification as set forth in the declaration *must be identical* under the Java rules." (A164, emphasis in original). The district court's finding applies with equal force to all programming languages.

The continued development of software, particularly in the cloud computing and mobile device markets, depends on the free use of APIs. In fact, developers have relied on the ability to use APIs and provide re-implementations as a source of innovation for over three decades. Oracle uses discredited "sweat of the brow"-type reasoning to argue that it should be entitled to monopolize the functional bits of code necessary for interoperability, contrary to the sound principles or policies of copyright law. What's more, similar attempts at aggregating control over APIs have been recognized by developers, the United States, and the European Union as dangerous to the app developer and cloud-computing industries. Copyright protection for declaring code would act as a tax on software development, leaving the public with lower-quality, less innovative software products.

As the district court's decision points out, the merger, short phrases, and method of operation doctrines all exclude the declaratory portions of APIs from copyright protection. The *scenes a faire* doctrine similarly precludes copyright protection for the declaratory elements of APIs that are dictated by practical realities like interoperability and file format compatibility.

Amici urge the Court to uphold the district court's decision and preserve the software community's ability to continue developing innovative, interoperable, open source, and cloud computing software through re-implementation of APIs.

ARGUMENT

Oracle and its amici want this Court to believe that, in the absence of copyright protection for API declaring code, the software industry will grind to a halt. What Oracle neglects to say is that for the past forty years, the software industry has experienced vibrant innovation despite the longstanding and universally-held belief among developers that declaring code was not protected by copyrights. In fact, many in the industry (in addition to the United States and European Union) see aggregation of control over APIs as a dangerous anti-competitive force. Permitting companies like Oracle to exercise near-total control over declaring code would severely limit Amici's ability to continue innovating, particularly in the API-dense industry niches of cloud computing and mobile devices.

I. APIs and their re-implementation are integral to continued innovation in the software industry.

This Court has been presented with several different iterations of what application programming interfaces, or APIs, mean in the specific context of this case. Oracle and some of its amici have defined APIs as "the vast array of Java programs to perform often-needed functions." (Oracle Br. 8; BSA Br. at 5). Another amicus brief defines APIs as software that "provides an interface that allows a user, computer, or piece of software to communicate with another piece of software." (Spafford, et al. Br. 8). It is true that, as their name suggests, application programming interfaces provide an interface between the inputs of one application and the outputs of another. But, as Microsoft acknowledges, most of the amici briefing the Court has received fails to address APIs “beyond the computer programs at issue here.” (Microsoft Br. 7). Understanding the extent to which the software industry depends on the free exchange and use of API declaring code is necessary to comprehend the chilling effect that Oracle’s position would have on software innovation and development.

APIs are the fundamental building blocks of software programs, used in every software interaction. APIs can be somewhat abstract, so it can help to think of a physical analogue—the Lego® brick. There are many different shapes and colors of Lego® bricks, but they all share one thing in common: the bumps on the top of each brick and the matching holes on the underside. These bumps and holes are the “interface” that allow different bricks to be joined together. A person building with Lego® bricks can join the bricks together in almost unlimited ways, using different colors and different shapes, but the interfaces must be all be compatible or the bricks simply won’t work together.

Software is similar. The “declaring” code in APIs is like the bumps on the Lego® bricks. This declaring code has various constraints due to the underlying nature of the system and the desire for compatibility. The “implementing” code is like the color and the shape of the brick—almost infinitely variable, the product of individual work and creativity.

APIs are the interfaces that allow hardware and software components to exchange information, to pass control, and to communicate with each other. They are ubiquitous and tend to be almost mechanical in nature. For example, the on-board computer system in a car contains APIs to interface between the input of a gas pedal and the output of a speedometer. When a driver puts pressure on a gas pedal to accelerate, the data from the gas pedal sensors is sent through an API to the speedometer, which generates an output in the form of a display of the miles per hour at which the car is traveling.[4]

APIs are vital to the entire software industry, including developers building apps for mobile devices, smartphones, and tablets. Companies wishing to attract developers frequently provide APIs that allow developers to share often-used functions without re-inventing the wheel. This explains why the most popular app development platforms—Google's Android platform and Facebook—are those that provide developers free access to robust APIs.

But APIs also make users' lives easier, allowing for file format compatibility and interoperability between devices and programs. An application is only able to function on a smartphone device if the application and smartphone operating

[4] For purposes of clarity, this example is slightly oversimplified. For example, when the car is in neutral, the speedometer will not change in response to pressure on the gas pedal. Nevertheless, the example is useful for understanding the functional nature of APIs.

systems share identical API declaring code. Identity of implementing code is not necessary to achieve interoperability.[5]

Likewise, APIs are integral to the cloud computing industry that Rackspace is pioneering. *Every* interaction over the Internet takes place using commonly-accepted standards for sending commands back and forth, returning responses, and sharing information—an API. Because cloud computing necessarily occurs without physical interaction between the cloud and the customer's computer system, all of the cloud computing software functionality is delivered to the customer via APIs. In other words, a user's mobile device communicates with the cloud entirely through APIs. As mobile devices and cloud computing become ever-larger parts of the software industry, APIs will continue to grow in importance to software developers.

A. API re-implementation is common in the industry and considered a legal way to obtain similar functionality and interoperability.

Oracle and its amici go to great lengths to portray their legal theory regarding declaring code's copyrightability as one deeply rooted in thirty years of copyright jurisprudence. But Oracle's position is completely contrary to well-established

[5] This point is an important one for two reasons. In the context of this case, Google copied the declaring code from the Java APIs at issue, but not the implementing code. In the broader context of the software industry, it is common practice to "re-implement" or "alternatively implement" APIs. This involves copying the declaring code of an API to maintain compatibility, but writing re-implementing code to achieve similar functionality and avoid copyright infringement.

industry practice. Before this case, no one in the software industry considered re-implementation of an API—that is, using the same declaring code but devising new implementing code—a violation of copyright law. In fact, long before Google re-implemented the Java APIs, programmers had re-implemented APIs to achieve similar functionality or interoperability. Software designers have long abided by Judge Alsup's conclusion that "so long as the specific code used to implement a method is different, anyone is free under the Copyright Act to write his or her own code to carry out exactly the same function or specification of any methods used" in an API. (A132).

Since the earliest days of software development, programmers have exercised their freedom to re-implement APIs to improve or compete with existing programs. For example, in the 1970s, developers at Bell Laboratories created the proprietary UNIX operating system. Because UNIX was so expensive, other programmers began re-implementing the UNIX APIs to develop less costly (or, in some cases, free) operating systems with similar functionality. The most famous and widely used of these clones, Linux, was developed in 1991. Linux borrowed heavily from the MINIX system, an earlier UNIX clone. When asked whether Linux had improperly copied MINIX APIs, the MINIX developer replied that while Linux copied the "layout of the file system" and "the names in the source tree," it had re-implemented the APIs.[6] In his eyes, and the eyes of most software developers, the Linux developer had committed no copyright violations. In fact, Linux is only one of a number of operating systems that have re-implemented the UNIX APIs, creating compatible operating systems that vary in their implementation and underlying functionality, but use the same interfaces to communicate between the different parts of the system. For example, even Microsoft has re-implemented the UNIX APIs in the "Interix" subsystem included with its server products.[7] This was done to provide application-level interoperability and allow Microsoft to better compete with UNIX-based systems.[8]

[6] Andrew S. Tanenbaum, *Some Notes on the "Who wrote Linux" Kerfuffle, Release 1.5*, (May 20, 2004), http://www.cs.vu.nl/~ast/brown/.

[7] Interix is an optional, POSIX-conformant Unix subsystem for Windows NT operating systems. Interix is a component of Windows Services for UNIX and a superset of the Microsoft POSIX subsystem. Like the POSIX subsystem, Interix is an environment subsystem for the NT kernel. It includes numerous open source utility software programs and libraries. *See* Stephen R. Walli, *INTERIX: UNIX: Application Portability to Windows NT via an Alternative Environment Subsystem*, http://stephesblog.blogs.com/papers/usenix-interix.pdf. Interix versions 5.2 and 6.0 are respective components of Microsoft Windows Server 2003 R2, Windows Vista Enterprise, Windows Vista Ultimate, and Windows Server 2008 as Subsystem for Unix-based Applications (SUA). Version 6.1 is included in Windows 7 (Enterprise and Ultimate editions) and in Windows Server 2008 R2 (all editions). *See* SERVICES FOR UNIX—INTEROPERABILITY BLOG, http://blogs.msdn.com/b/sfu/ (last visited May 29, 2013).

[8] "Application source code portability is one of the cornerstones of most open systems definitions. The intention is that if an application is written to a particular model of source-code portability, it can port relatively easily to any platform that supports the portability model." Walli, *supra* note 6 at 1.

Another free, open source[9] project called Wine has been re-implementing Microsoft's Windows APIs since 1993.[10] The project translates the declaring code of the Windows APIs so that Windows-based applications can run on other operating systems like Mac OSX and Linux.[11] Wine developers go to great lengths to ensure that the implementing code of their APIs is developed without reverse engineering or access to any Microsoft implementing code in an effort to avoid copyright infringement.[12]

Re-implementing APIs for the purpose of competing with an existing product is a regular and accepted business practice. In the early days of word processing, WordPerfect was the dominant word processing program, even ahead of Microsoft Word. Within each WordPerfect document were a series of codes—instructions to the word processing program that described the layout and formatting of the document. In order to compete, Microsoft wrote its own implementing code that

[9] Open source software is, generally speaking, software in which the source code is made publicly available. OPEN SOURCE INITIATIVE, http://opensource.org/osd (last visited May 29, 2013). The open source movement rests on the theory that freely-available code harnesses the development skills of more programmers and increases peer review of code, leading to better quality and lower-cost software.

[10] Wine HQ, *Explaining the Wine Project* http://www.winehq.org/about/ (May 29, 2013).

[11] Wine HQ, *Wine User Guide, Chapter 1*, http://www.winehq.org/docs/wineusr-guide/what-is-wine (last visited May 29, 2013).

[12] CodeWeavers, Inc., *CodeWeavers and Microsoft Licensing Questions*, http://www.codeweavers.com/products/faq/licensing/ (last visited May 29, 2013).

allowed Microsoft Word to open and modify WordPerfect files. In order for Word to open a WordPerfect document containing a bolded word and maintain WordPerfect's original bolded formatting, Word must 'understand' the code WordPerfect uses to designate bolded text. Word does this by utilizing WordPerfect's pre-set definitions (analogous to declaring code) for bolded words and translating them into Word implementing code. In this example, Microsoft has used WordPerfect code only to the extent necessary to allow users to open different file formats in Microsoft Word and therefore achieve some level of file format compatibility.

These examples demonstrate the universal understanding of the unprotected status of declaring code and the complete lack of historical support for Oracle's argument. Almost any time a software product advertises that it is "compatible" with a competitor, developers have written new implementing code and kept the external interfaces—the APIs—the same. Reversing the district court would disrupt these businesses, prejudice the large community of app and app platform developers that rely on compatibility between competitors, and put an end to innovative open source projects like Linux and Wine.

B. Aggregated control over APIs is a dangerous anti-competitive force.

Allowing Oracle, or any party, exclusive copyright-based controls over APIs would have a significant anti-competitive effect on the industry. In fact, as part of

Oracle's purchase of Sun Microsystems (the acquisition that led to Oracle's ownership of the Java APIs at issue here),[13] Oracle also acquired an open source database management system called MySQL. Upon investigating the proposed merger between Oracle and MySQL, the European Commission on Competition expressed concerns that Oracle would discontinue or privatize the open source database, thereby eliminating some of its competition.[14] The Commission was particularly worried that permitting Oracle to control the MySQL APIs would reduce the choices available to consumers and increase prices of database software. In order to quell the European Commission's worries about the anti-competitive potential of the merger, Oracle released a public statement promising that the MySQL APIs would remain publicly available for use.[15] Based in part on these promises, the European Commission approved the merger.

Microsoft, which filed an amicus brief supporting Oracle, has also tried to leverage API access to stifle competition only to ultimately have to back away from its position. In 1998, the United States brought a civil anti-trust suit against Microsoft, alleging that the software giant had improperly manipulated the internet

[13] Oracle Br. 2, n1.

[14] Press Release, Europa, Mergers: Commission Opens In-Depth Investigation Into Proposed Takeover of Sun Microsystems by Oracle (Sept. 3, 2009), http://europa.eu/rapid/press-release_IP-09-1271_en.htm?locale=en.

[15] Press Release, Oracle Makes Commitments to Customers, Developers and Users of MySQL (Dec. 14, 2009), http://www.oracle.com /us/corporate/press/042364.

browser market to favor its Internet Explorer program.[16] Among the many allegations in that suit, the United States asserted that Microsoft had altered its Windows APIs to favor interoperability with its Internet Explorer web browser as a way to shut out competition from other browser manufacturers. As part of the settlement Microsoft reached with the United States, Microsoft was required to publicly release its APIs for the "purpose of interoperating with a Windows Operating System Product." In other words, Microsoft was forced to make its APIs publicly available so that other developers could utilize the declaring code for products, including web browsers, that were fully compatible with the Windows operating system. In the judgment of the United States Department of Justice, free access to the declaring code of Windows APIs was necessary to maintain adequate competition in the software industry.

Oracle seeks to achieve through the Copyright Act what the European Commission and United States Department of Justice determined that Oracle and Microsoft should not be permitted to achieve through mergers and anti-competitive business practices. The result that Oracle seeks in this case—the ability to control who can use the declaring code of the Java APIs and exact a tax from each developer who seeks to use the APIs—would create a similarly anti-competitive force in the

[16] *See* Complaint, *United States v. Microsoft*, Case No. 98-1232 (Dist. D.C.), available at http://www.justice.gov/atr/cases/f1700/1763.htm.

software industry.[17] The result would also jeopardize the livelihood of the millions of individuals employed as app developers or employed by publishers and platforms.

C. Copyright protection of declaring code is not necessary to incentivize software innovation.

Activity in areas like cloud computing, open source software development, and the mobile device market demonstrates that robust protection for the declaring code of APIs is unnecessary to spur software innovation. As described above, software developers have always operated on the assumption that re-implementation of APIs, and therefore the use of declaring code, was not a violation of copyright law. Obviously, the software industry has been quite innovative so far. Despite Oracle's protestations, the software industry has thrived for the last forty years without copyright protection for the declaring code of APIs.

In fact, open source development is increasingly being recognized as a driving force of software innovation.[18] The philosophy behind open source software is that

[17] For further commentary on the relationship between compatibility, competition, and innovation in the marketplace, *see* Timothy S. Teter, Note, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 Stan. L. Rev. 1061, 1066-1070 (1993).

[18] Matt Asay, *Open Source Gains While Proprietary Software Declines*, CNET (Apr. 20, 2009, 10:07 a.m. PDT), http://news.cnet.com/8301-13505_3-10223005-16.html (data indicating that the percentage of organizations using proprietary software was expected to dip, while the percentage using open source software was expected to rise).

open access to APIs fosters more creative development and more peer review of code, yielding higher quality, more innovative software. Rackspace's open source OpenStack project, founded with help from NASA, is on the cutting edge of cloud technology. According to the OpenStack website, the "open development model is the only way to foster badly-needed cloud standards, remove the fear of proprietary lock-in for cloud customers, and create a large ecosystem that spans cloud providers."[19] The popularity of open source development—and the fact that for-profit companies are investing in open source projects—alone demonstrates that the ability to control who re-implements APIs is not necessary to incentivize software innovation.

But a recent example serves to underscore the point. In early 2010, Google released a microblogging program called Google Buzz but did not immediately make the underlying APIs available to developers.[20] Because developers were not able to immediately make compatible software or to integrate Google Buzz into their existing sites and platforms, the program failed. Google discontinued Buzz in late

[19] Openstack Cloud Software, *What is OpenStack?*, http://www.openstack.org/ (last visited May 29, 2013).

[20] CloudAve, *PaaS is the Future of Cloud Services: APIs are the Key*, http://www.cloudave.com/282/paas-is-the-future-of-cloud-services-apis-are-the-key/ (last visited May 29, 2013) ("They launched Google Buzz and Google Latitude with much fanfare but they didn't release the API for these services. This resulted in lack of interest among the developers and, hence, failed to gain traction among the users.").

2011.[21] In contrast, Google made the Android APIs open and available, and innovation for the operating system exploded.[22] Android now claims to be the most popular mobile device operating system world-wide.[23] Facebook similarly permits developers unlimited access to its APIs and is also one of the largest and most robust app development platforms. Google's experience demonstrates that the ability to withhold developers' use of APIs is not only unnecessary for innovation, it may actually hamper innovation.

Oracle and its amici claim that, without copyright protection for declaring code, the software industry will collapse. This claim is not supported either by historical industry practice or current trends in software development. In fact, the opposite is true. Permitting Oracle and others to determine who can use APIs and for what purpose would undermine innovative software development programs and stifle developers' ability to create compatible and interoperable software applications.

[21] Google, *Google Buzz Has Gone Away, But Your Posts Are Yours to Keep*, https://support.google.com/mail/answer/1698228?hl=en (last visited May 29, 2013).

[22] Open Handset Alliance, *Android FAQ*, http://www.openhandsetalliance.com/android_overview.html (last visited May 29, 2013).

[23] ANDROID, http://www.android.com/about/ (last visited May 29, 2013).

II. Long-standing doctrines of copyrightability are already calibrated to balance industry needs.

Oracle and its amici suggest that any balancing between the rights of an original author and those of a subsequent innovator should be done using the infringement prong of copyright analysis, particularly the doctrine of fair use. But the doctrine of fair use is a slippery one, once referred to as "the most troublesome in the whole law of copyright." *Universal City Studios v. Sony Corp. of Am.*, 659 F.2d 969 (9th Cir. 1981). The doctrine requires careful balancing of four nuanced, non-exclusive factors to determine whether an individual's use—while technically within the scope of infringement—should be considered unlawful. *See* 17 U.S.C. § 107. The statute does not set forth how much weight a court should give to any of the four factors or whether any of the individual factors is sufficient for a finding of fair use.

Were this Court to accept Oracle's position, almost every player in the industry would be susceptible to suits for copyright infringement when using declaring code. If liability for the entire market were determined based on a case-by-case determination of fair use (an already unpredictable doctrine), developers would be unable to adequately predict their exposure. This would significantly increase the transaction costs associated with creating interoperable software, including all uses of cloud computing, which is provided via APIs.[24] This presents an untenable burden for Amici, the thousands of developers building on our platforms, and the hundreds of thousands of customers that use APIs to interact with our products.

It is possible, as evidenced by the district court's decision in this case, to adequately balance the rights of copyright holders, developers, and competitors through the existing doctrines regarding merger, short phrases, and methods of operation. It is also possible to adequately balance these rights through the *scenes a faire* doctrine.[25] All of these doctrines would provide more predictable results for those in the industry and therefore achieve more cost-efficient innovation than the fair use doctrine.

The *scenes a faire* doctrine prohibits copyright protection "when external factors constrain the choice of expressive vehicle." *Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004). While more traditionally applied to the literary context, *scenes a faire* in the software context prohibits

[24] *See also* Jacqueline D. Lipton, *IP's Problem Child: Shifting the Paradigms for Software Protection*, 58 Hastings L.J. 205, 233 (2006) ("It might be much more useful for the continued development of the software industry if there were some ex ante test for copyrightability of code. . .").

[25] Some circuits consider both merger and *scenes a faire* to be doctrines of infringement which should be applied to the substantial similarity analysis. Others consider the doctrine one of copyrightability. *See Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 559 (6th Cir. 2004) (Feikens, J. dissenting) (cataloging circuit split).

copyright protection on those elements of a program "dictated by practical realities"—like the interoperability and file format compatibility concerns associated with the copying of API declaring code. *Id.* Commentators have long suggested the use of *scenes a faire* and the related doctrine of merger to remove code necessary for compatibility from the scope of copyright protection.[26] Courts have also indicated that *scenes a faire* could be a useful tool in trimming copyright protection for elements of code dictated by compatibility concerns.[27] The district court in this case observed that *scenes a faire* likely barred protection on the method header and names Oracle sought to protect. (A165-66, n.9).

Case law, commentary, and common sense all demonstrate that copyright protection for declaring code is unnecessary and potentially harmful to the software industry. While fair use analysis may ultimately reach the same result on a case-by-case basis, the *scenes a faire* doctrine, in addition to those invoked by the

[26] *See* Teter, *supra* note 17 at 1062-63, 1097 (explaining how copyright protection on code required for compatibility "may enable a software producer to achieve a far-reaching monopoly" and "affect[] innovation" and acknowledging that while "elements necessary for compatibility warrant some form of intellectual property protection, one must acknowledge that copyright is unsuited for the task"); Andrew B. Hebl, Note, *A Heavy Burden: Proper Application of Copyright's Merger and* Scenes A Faire *Doctrines*, 8 Wake Forest Intell. Prop. L.J. 128 (2007).

[27] *See Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823 (10th Cir. 1993) (noting that the application of *scenes a faire* to 'interfacing' "has the potential to effect widely the law of computer copyright") (citing several other cases which apply *scenes a faire*); *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1525-27 (9th Cir. 1993).

district court, can serve as a more cost-efficient method for balancing the rights of follow-on innovators.

CONCLUSION

For the above reasons, Amici urge the Court to affirm the district court's decision and refuse Oracle the ability to expand copyright into anti-competitive, over-broad protection on API declaring code so fundamental to continued innovation in the software industry. By finding that code necessary for functional compatibility—like the declaring code of APIs—is an unprotectable idea, either under the district court's analysis or the doctrine of *scenes a faire*, this Court can protect the open source and interoperability projects that are vital to continued progress in the industry. At its core, protection of innovation is the purpose the Copyright Act was meant to fulfill.

Dated: May 30, 2013

Respectfully submitted,

Chad Ruback
The Ruback Law Firm
8117 Preston Road, Suite 300
Dallas, Texas 75225
(214) 522-4243
(214) 522-2191 fax

/s/ Chad Ruback
Chad Ruback

Attorney for Amici Curiae Rackspace US, Inc., Application Developers Alliance, TMSOFT, LLC, and Stack Exchange Inc.

ECF CERTIFICATION

I hereby certify (i) the required privacy redactions have been made pursuant to Federal Rule of Civil Procedure 5.2; (ii) the electronic submission is an exact copy of the paper document; (iii) the document has been scanned for viruses using Symantec Endpoint Protection active scan and is free of viruses; and (iv) the paper document will be maintained for three years after the mandate or order closing the case issues.

/s/ Chad Ruback
Chad Ruback

CERTIFICATE OF COMPLIANCE

In accordance with Fed. R. App. Proc. 32(a)(7)(C), the undersigned certifies that this brief complies with the type-volume limitations of Fed. R. App. Proc. 32(a)(7)(B).

Excluding the exempt portions of the brief (as provided in Fed. R. App. Proc. 32(a)(7)(B)(iii) and Fed. Cir. R. 32(b)), this brief includes 5,164 words.

This brief has been prepared in a proportionally spaced typeface using Microsoft Word 2010 in 14 point Times New Roman font. As permitted by Fed. R. App. Proc. 32(a)(7)(C), the undersigned has relied upon the word count of this word-processing system in preparing this certification.

/s/ Chad Ruback
Chad Ruback

Dated: May 30, 2013

Form 30

FORM 30. Certificate of Service

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

CERTIFICATE OF SERVICE

I certify that I served a copy on counsel of record on 5/30/13
by:

□ US mail
□ Fax
□ Hand
☒ Electronic Means
(by email or CM/ECF)

Chad M. Ruback
Name of Counsel

/s/ Chad M. Ruback
Signature of Counsel

Law Firm The Ruback Law Firm

Address 8117 Preston Road, Suite 300

City, State, ZIP Dallas, Texas 75225

Telephone Number (214) 522-4243

FAX Number (214) 522-2191

E-mail Address chad@appeal.pro

NOTE: For attorneys filing documents electronically, the name of the filer under whose log-in and password a document is submitted must be preceded by an “/s/” and typed in the space where the signature would otherwise appear. Graphic and other electronic signatures are discouraged.