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-3561, Judge William H. Alsup.

CORRECTED BRIEF OF *AMICI CURIAE* OF SOFTWARE INNOVATORS, START-UPS, AND INVESTORS IN SUPPORT OF AFFIRMANCE

JENNIFER M. URBAN, ESQ.
Counsel of Record
SAMUELSON LAW, TECHNOLOGY
AND PUBLIC POLICY CLINIC
U.C. BERKELEY SCHOOL OF LAW
585 Simon Hall
Berkeley, California 94720-7200
(510) 642-7338 Telephone
(510) 643-4625 Facsimile

Attorney for Amici Curiae

June 7, 2013

***Amici Curiae* Software Innovators, Start-ups, and Investors:**

Apiary, Inc.

Azavea, Inc.

Daniel Bricklin

Bright Funds, Inc.

Esther Dyson

EDventure Holdings

Engine Advocacy

fasterlighterbetter d/b/a Copper

Foundry Group

Robert Glushko

Hattery

Mitchell Kapor

Katkar Flink Corporation d/b/a Hipiti

Mozilla Corporation

Tim O'Reilly

Raymond Ozzie

Union Square Ventures, LLC

Zubhium, Inc. d/b/a Vessel

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rules 29(a) and 47.4, counsel for *Amici Curiae*

certifies that:

1.   The full names of every party or *amicus* represented in the case by me are:

See attachment to Certificate of Interest.

2.   The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

N/A

3.   All parent corporations and any publicly held companies that own 10 percent or more of stock of any party or *amicus curiae* represented by me are:

Mozilla Foundation is the parent corporation of Mozilla Corporation.

4.   The names of all law firms and the partners or associates that appeared for a party or *amicus* now represented by me in the trial court or are expected to appear in this Court are:

Jennifer M. Urban
Samuelson Law, Technology and Public Policy Clinic
U.C. Berkeley School of Law

Signed: /s/ Jennifer M. Urban   Dated: May 31, 2013
                        Jennifer M. Urban
                        Counsel of Record
                        Attorney for Amici Curiae

ATTACHMENT TO CERTIFICATE OF INTEREST
LIST OF AMICI CURIAE
(In alphabetical order – descriptions available at
http://www.law.berkeley.edu/amici.htm)

1. Apiary, Inc.

2. Azavea, Inc.

3. Daniel Bricklin

4. Bright Funds, Inc.

5. Esther Dyson

6. EDventure Holdings

7. Engine Advocacy

8. fasterlighterbetter d/b/a Copper

9. Foundry Group

10. Robert Glushko

11. Hattery

12. Mitchell Kapor

13. Katkar Flink Corporation d/b/a Hipiti

14. Mozilla Corporation

15. Tim O'Reilly

16. Raymond Ozzie

17. Union Square Ventures, LLC

18. Zubhium, Inc. d/b/a Vessel

TABLE OF CONTENTS

TABLE OF AUTHORITIES

News, Periodicals, and Journal Articles:

Web Pages:

INTEREST OF *AMICI* [1]

Amici are software innovators, start-ups, and investors. The signatories on this brief include innovators, and founders of software and Internet companies that actively innovate in and compete across a wide array of markets. Signatories also include investors who invest in, and are expert in assessing the risks of investing in, companies that rely on APIs and other interoperability tools. *Amici* have broad first-hand experience in the role of interoperability—and the balanced and stable copyright rules on which it depends—in driving innovation in the technology sector. A full list of *amici* with individual descriptions can be found at http://www.law.berkeley.edu/amici.htm.

Amici's shared interest in this case is in preserving the deliberate balance Congress and the courts have established for software copyright, including longstanding limits on copyrightability that enable innovation by fostering interoperability and competition. *Amici* join to explain the importance to innovation and investment in innovation of upholding the District Court's careful application of these limitations to the Java API elements at issue in this case.

---

[1] Pursuant to Fed. R. App. P. 29(a), all parties have consented to the filing of this brief. No party's counsel authored this brief in whole or in part. No person (or party) other than *amici*, their members, or their counsel contributed money that was intended to fund preparing or submitting this brief. University of California, Berkeley law students Christopher Civil and Michael Liu Su assisted in the preparation of this brief. Web sites cited in this brief were last visited in May 2013.

SUMMARY OF ARGUMENT

Interoperability between programs and systems is key to software innovation. It allows systems to connect and individuals to apply knowledge and skills from one environment to another. By easing the way for software developers to build upon existing platforms, interoperability allows efficient software ecosystems to grow, fueling the development of innovative new products and services and increasing competition to the benefit of consumers.

Copyright's foundational rules—including its carefully crafted limitations on copyrightability—are fundamental to the efficient development of interoperable systems. At least since *Baker v. Selden* was decided in 1879, it has been firmly established that copyright law extends only to creative expression and not to ideas. 101 U.S. 99 (1879). Similarly, copyright protection does not extend to methods of operation, procedures, processes, or systems. 17 U.S.C. § 102(b). Because they preserve the availability of the basic building blocks required to create communication between systems, these limitations are essential to ensuring that interoperability remains a reliable feature of software development. In turn, the resulting software ecosystems and platforms undergird rapid innovation in beneficial new products and services.

Regardless of the mix of intellectual property rights on which they choose to rely, innovators and their investors need certainty about how copyright applies to

computer programs. In this case, the district court properly ruled that the elements

of the Java APIs at issue are not copyrightable. It correctly recognized that these

elements instead constitute "methods of operation" that allow developers to access

functions within the Java programming language itself. Under this ruling,

computer programs remain covered by the same carefully crafted copyright

protection that has been in place for the past thirty years. Oracle's preferred

analysis, Opening Br. and Addendum of Pl.-Appellant 31–34 ("Appellant's Br."),

if adopted whole or in part, would upset this stable regime and drastically expand

copyrightability. This would not only limit innovators' ability to develop

interoperable systems—it would also introduce uncertainty into longstanding

software development practices and investment in software innovation.

Maintaining the conditions—including stable and balanced copyright

rules—that enable software innovation also means maintaining strongly

interoperable systems and competitive markets. The case at hand is a case in point.

The number of start-up software companies has dramatically increased over the

last decade as it has become comparatively easier to transform an idea into reality,

in significant part because of the interoperability fueled by APIs. By creating a

communicative interface between two software components,[2] APIs allow start-ups

to create programs that can communicate with and integrate the technology of

---

[2] See James Gosling et al., *The Java Application Programming Interface, Volume 1*, xv–xx (1996).

existing systems, powering development and increasing the programs' usefulness. APIs therefore support rapid innovation and afford start-ups a greater chance at success, enabling angel investors, venture capitalists, and other investors to readily fund innovative ideas and help bring them to market.

Relatedly, interoperability is essential to maintaining robust competition in software development. Because it allows individuals to efficiently build programs and services that can communicate with and complement existing products, interoperability supports market entrance by new competitors. This accelerates the development of new uses, products, and services in a wide variety of fields, including both traditional and open source development, fueling competition.

Reversing the district court's ruling would create uncertainty for software development that relies on APIs and development that feeds into interoperable systems more generally. This would harm what is presently a vital and robust industry, chilling both innovation and the investment that supports it. *Amici* therefore respectfully request that this Court affirm the district court's decision in order to ensure that copyright remains in proper balance, supporting interoperability instead of restricting it, and allowing innovation to flourish.

ARGUMENT

I.   Longstanding Limitations on Copyrightability Protect Software Innovation by Start-up Companies.

A.   Start-up Software Companies Are Key Drivers of Innovation.

Start-up companies are vital to the American economy as a source of jobs,

capital, and innovation. *See* Tim Kane, Ewing Marion Kauffman Foundation, *The*

Importance of Startups in Job Creation and Job Destruction at 3 (2010)

(describing the strong source of net job growth new companies provide).[3] Indeed,

American start-ups are the most innovative in the world, allowing them to fuel

American markets both home and abroad. *See* World Economic Forum,

Technology Pioneers 2013 at 9 (2013) (listing 13 American start-ups out of a total

of 23 as the most innovative in the world).[4]

These companies are characterized by their ability to quickly turn an

innovative idea into a product that benefits society. They have a storied history as a

robust source of American innovation: economic powerhouses such as Apple,

Microsoft, Google, Yahoo!, Intel, and Oracle itself all began as small start-ups.

Today's start-ups, including companies funded by *amici* and *amici* start-ups

themselves, continue to innovate new products and services that benefit every

---

[3] *Available at* www.kauffman.org/uploadedFiles/firm_formation_importance_of_startups.pdf.
[4] *Available at* www3.weforum.org/docs/WEF_TP_PushingNewFrontiers_Report_2013.pdf.

sector of society. For example, *amicus* Copper has created a new way to compensate online content creators and *amicus* Bright Funds, simplified tools to donate to charities online. *Amicus* Foundry supports companies engaged in a wide array of sectors: Brightleaf offers automated legal documentation, MakerBot builds and sells 3D printers and scanners, PivotDesk connects start-ups with office space, and Sympoz provides an online learning community. *Amicus* Apiary helps other companies use APIs to provide a multitude of services.

Start-ups and their investors rely on copyright's default rules in order to innovate these new software products and services. As described further below, longstanding limitations on copyrightable subject matter, such as disallowing copyright protection for ideas, systems, and methods of operation, and limiting the protection of "functional elements essential for interoperability," provide foundational support for the innovation process. *See Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 984–998 (N.D. Cal. 2012).

B. Start-up Innovation Depends on Interoperability Between Systems and Programs.

Importantly, copyright's traditional limitations support robust interoperability. Innovation in the software community flourishes because of interoperability. *See* Urs Gasser & John Palfrey, *Interop: The Promise and Perils*

of Highly Connected Systems 111-25 (2012) (finding that interoperability supports

innovation in the context of information and communication technologies).

Indeed, interoperability—whether created through controlled or "open"

ecosystems—is a foundational requirement of software innovation. To function,

software programs must be able to interact with other computer software and

hardware. Interoperability also allows customizations which connect to original

software, expanding the software's usefulness and its competitive value. *See id.* at

118. APIs, specifically, create interoperability between programs and systems,

giving start-up innovators the ability to connect to and build on what has come

before. APIs facilitate this interaction between programs because they "provide

access to functionality that is not easily achievable without the APIs . . . ." Jeffery

Stylos & Brad Myers, *Mapping the Space of API Design Decisions,* 2007 IEEE

Symposium on Visual Languages and Human-Centric Computing 51 (2007)

(hereinafter "IEEE Report").[5] In fact, programmers often "must use the provided

APIs because the implementation details are intentionally hidden" to protect

intellectual property rights in protectable elements of interacting programs. *Id.*

---

[5] *Available at* repository.cmu.edu/cgi/viewcontent.cgi?article=1168&context=hcii.

C.   Appropriate Limits on the Copyrightability of API Features Foster Interoperability, Fueling Innovation.

Software developers thus depend on APIs as a key method for achieving interoperability. The practice of creating via API-enabled platforms in turn depends on established limits to copyright protection. Indeed, start-ups, including many founded or funded by *amici*, can efficiently use APIs precisely because elements necessary to creating interoperability are not covered by copyright. The importance of these APIs to innovative software start-ups reflects the overall importance of preserving these limitations in copyright law.

1.   Start-up Software Development Relies on the Use of APIs.

Start-ups depend on APIs to efficiently bring new ideas to the market. One study has found that software programs that implement APIs decrease the amount of time they take to bring products to market by 30%. *See* Fern Halper, Judith Hurwitz, & Marcia Kaufman, *A Web API Study: The Benefits of APIs in the App Economy* (2011).[6] Indeed, there has been a rapid expansion of API service providers like *amicus* Apiary, which provides a platform, or "blueprint," for developers to quickly describe, document, test, and implement new APIs.

---

[6] *Available at* www.hurwitz.com/index.php/recent-research/cloud-computing/doc_details/128-web-api-study-the-benefits-of-apis-in-the-app-economy.

Start-ups[7] depend on APIs in order to perform tasks that they otherwise would not be able to accomplish. For example, a start-up that integrates Twitter or Facebook services so that its users can communicate with their connections must use an API. Similarly, a start-up that allows its users to use a PayPal account to pay for goods and services must use an API. *See* IEEE Report at 51. Start-ups also use APIs because they contain the phrases and structures of code necessary to efficiently build innovative new products. In this vein, APIs increase efficiency by enabling programmers to build from existing code instead of "writing it from scratch." *See* IEEE Report at 53–54; *see also* Br. *Amici Curiae* Computer Scientists in Supp. of Def.-Cross Appellant and Affirmance.

APIs vary greatly in available features, allowing innovators to connect with existing systems to provide an enormous range of services. A very small sampling includes mobile payments,[8] online discussion,[9] web search,[10] project management,[11] banking,[12] motion tracking,[13] music,[14] and information exchanges between mobile devices.[15] By using APIs, start-ups can also provide valuable

---

[7] This is true of all software developers. *See* IEEE Report at 51.
[8] *See* Copper Inc., http://www.copper.is.
[9] *See* Disqus, http://www.disqus.com.
[10] *See* OpenSearch, http://www.opensearch.org.
[11] *See* Basecamp (37signals, LLC), http://basecamp.com.
[12] *See* Simple Finance Technology Corp., http://www.simple.com.
[13] *See* Microsoft Kinect, http://www.microsoft.com/en-us/kinectforwindows/.
[14] *See* Last.fm LTD, http://www.last.fm.
[15] *See* Bump Technologies, Inc., http://bu.mp.

enhancements that they otherwise could not offer consumers. For example, start-ups can use the Google Maps API to incorporate Google Maps into their applications. WallIt,[16] for example, uses the Google Maps API to access location services and Google Street View. Without APIs like this one, most start-ups would likely be unable to include map features in their products, due to the complexity of the undertaking: Google needs over 7100 employees to build Google Maps. *See* Nicholas Carlson, *Apple Has ~7,000 Fewer People Working On Maps Than Google,* Business Insider (Sept. 21, 2012).[17] The availability of APIs for labor-intensive products Google Maps thus fuels innovators' ability to complement the existing product and usefully enhance their own products.

Beyond these benefits, however, the availability of APIs is simply central to contemporary software innovation and program operation. *See* Doreen Bloch, *4 Reasons to Develop an API*, Bootstrappist (April 9, 2012) ("No website, network or app is isolated . . . . Websites "talk" to one another through . . . APIs, sets of rules that developers use to facilitate communications among different platforms to share data structures, protocols, and more.")[18]; Adam DuVander, *7,000 APIs: Twice as Many as This Time Last Year*, ProgrammableWeb (Aug. 23, 2012) ("If an

---

[16] See WallIt, http://launch.wallitapp.com.
[17] *Available at* http://www.businessinsider.com/apple-has-7000-fewer-people-working-on-maps-than-google-2012-9#ixzz2O8dTBdXy.
[18] *Available at* http://www.bootstrappist.com/archives/4-reasons-to-develop-an-api/.

. . . app does anything interesting, it likely needs . . . an API.").[19] The extent to

which start-ups depend on APIs is exemplified by the fact that every start-up

supported by *amici* Hattery and Foundry use at least one API.

Unsurprisingly, the number of available APIs is growing at a rapid pace,

with hundreds of new APIs created every month. *See* DuVander, *supra*. One

popular central listing currently indexes over 9100 APIs—a number that has grown

from 7600 in just the seven months from October 2012 to May 2013. *See*

ProgrammableWeb API Directory.[20]

> **2.   Reversing the District Court's Ruling Would Introduce
> Legal Uncertainty to the Use of APIs, Harming Start-ups'
> Ability to Cumulatively Innovate.**

Were Oracle's preferred rule adopted, API-fueled innovation—and indeed

all software-based innovation—would be threatened. Start-ups depend on the fact

that copyright is tailored to encourage cumulative innovation. The law "withholds

copyright protection from creative building blocks . . . to stimulate the production

of the most abundant possible array of . . . expression." Paul Goldstein, Goldstein

on Copyright, § 2.3.1.1.

---

[19] *Available at* http://blog.programmableweb.com/2012/08/23/7000-apis-twice-as-many-as-this-time-last-year/.

[20] *Available at* http://www.programmableweb.com/apis/directory. Because an API can be created by anyone, with no need to catalogue its existence, the actual number of APIs is likely much higher than what is tracked in such indexes.

Start-ups following the API-supported model of development described above—and indeed all innovators—depend on the availability of these building blocks in their practice. Dramatically expanding copyright's scope, as Oracle seeks to do, would upend this balance. If the elements of the Java API at issue here were held to be copyrightable, this would create uncertainty regarding other API elements long understood to be free from copyright protection. New copyright protections—or uncertainty regarding the default rules of copyrightability—could be used as a weapon to block innovation. Further, companies that presently provide complementary services by connecting to existing systems would be required instead to undertake a development process fully duplicative of the existing system, limiting offerings to consumers.

Faced with such uncertainty, start-ups may hold off from developing and using APIs, resulting in a significant loss in innovation. As discussed further in the next section, this concern is heightened by the fact that start-ups are especially sensitive to the economic costs of such uncertainty and its effects on their ability to generate funding.

> **3. Reversing the District Court's Ruling Would Chill Investment and Harm Start-ups' Ability to Obtain Funding.**

Start-ups need money to turn ideas into reality. Angel, venture capital, and other investors are important sources of this funding and play a significant role in

bringing new innovations to society. *See* Samuel Kortum & Josh Lerner, *Assessing the Contribution of Venture Capital to Innovation*, RAND Journal of Economics (2000) (Finding that increases in venture capital funding in a sector are associated with statistically significant higher rates of innovation). In 2012 alone, venture capital firms made over $4 billion of initial investments in 1,174 companies, largely at the crucial "seed- and early-stage" of their development paths. National Venture Capital Association, Yearbook 2013 at 13–14 (2012) (out of a total of $26.7 billion invested in 3,143 companies)[21]; *see also* Darian M. Ibrahim, *The (Not So) Puzzling Behavior of Angel Investors*, 61 Vand. L. Rev. 1405, 1407 (2008) (discussing the boosts to employment and gross domestic product that investor-backed firms have provided in the 2000s).

Investors like *amici* are more likely to fund start-ups that can rely on APIs. An idea that can make use of already-developed features requires less initial footwork and thus stands a higher chance of coming to market. A start-up that uses APIs can come to market on as little as a few hundred dollars, compared to the millions of dollars required for conventional software development. Moreover, start-ups that use APIs benefit from a greater ability to connect with existing markets. By making it possible for new products to connect to existing software

---

[21] *Available at* http://www.nvca.org/index.php?option=com_content&view=article&id=257&Itemid=103.

ecosystems, APIs reduce the time it takes to bring an innovative new idea to the market. These efficiencies translate into lower business risk and make successful market entry more likely. Investors are thus more willing to fund software start-ups that can use APIs; this is especially true for early-stage investments.

Should the fundamental copyrightability rules related to APIs shift as Oracle requests, however, software developers and their investors would face a new source of uncertainty and litigation threats. This would harm investment in start-ups that use APIs: investors are much less likely to invest if they fear that their investments will go towards copyright litigation. *See* Michael A. Carrier, *Copyright and Innovation: The Untold Story*, 2012 Wis. L. Rev. 891, 916 (2012) (describing how past copyright infringement threats have created "wastelands" of technology fields due to lack of venture capital funding). For example, investors backed away from the entire field of technology involving online music downloading following a small but significant number of adverse copyright infringement cases. *See id.* Further, these negative effects would not be limited to investment in API-fueled software innovation. Because Oracle's preferred interpretations, Appellant's Br. at 31-34, would confuse well-settled law regarding protectable and unprotectable elements of computer programs more generally, see *infra* Section III, the uncertainty would also extend more generally into investment in computer software start-ups.

14

Destabilizing copyright limitations would thus also destabilize funding

sources for software start-ups, disrupting their ability to bring innovative new

products and services to market.

II.   Longstanding Limitations on Copyrightability Protect and Encourage Competition in Software Innovation.

Destabilizing copyright limitations would also undercut the competitive

benefits software innovators bring to markets. Proper limitations on copyrightable

subject matter are necessary for competition because they ensure that the facts,

ideas, and other essential elements underlying copyrighted works are equally

available to new market entrants and incumbents. *See generally* Ariel Katz,

Copyright and Competition Policy, forthcoming in *Handbook of the Digital*

Creative Economy (Christian Handke and Ruth Towse, eds. 2013).[22]

The scope of copyright in computer programs has practical effects on

competition because participation in these networks requires interoperability:

certain elements of the communicated information must be the same on both sides

of the exchange. The Java API aspects at issue here comprise such elements and

functions. As discussed below, these elements have long been considered

uncopyrightable. Expanding copyright as Oracle requests would dramatically

increase incumbents' ability to use copyright to limit competition by preventing

---

[22] *Available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2231811.

15

newcomers from accessing established systems and from innovating on top of older ideas.

This poses a clear and recognized threat to open competition within markets. *See, e.g.*, *Who Owns the Perk in Java?*, The Economist Blog (May 8, 2012) (noting that "many tech types are jittery about a verdict fully in favour of Oracle" because "[e]quivalent API functions based on distinct source code abound across all aspects of hardware, software and services, on the [I]nternet and offline" and cautioning that accepting Oracle's argument could "reshape the nature of technological development.").[23]

A. Appropriate Limitations on Copyrightability Are Essential to Competitive Markets.

1. Clear and Reliable Copyrightability Limitations Allow Market Entrants to Participate Effectively in Existing Networks and Innovate on Top of Existing Platforms.

Were Oracle to succeed in its request to expand copyrightability, the ensuing damage to interoperability would represent a serious setback to market entrants' ability to replace or compete alongside existing products. *See* Jonathan Band, *Interfaces on Trial 2.0* 1-5 (2011) (explaining that interoperability enables both communication between platforms and replacement of products by competitors).

---

[23] *Available at* www.economist.com/blogs/babbage/2012/05/oracle-v-google.

To achieve a truly competitive market, software programs must be able to communicate effectively with other programs. Copyright law reflects this understanding in the limits to its protection for software. In *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 709-10 (2d Cir. 1992), the Second Circuit stated that program elements dictated by factors such as "compatibility requirements of other programs with which a program is designed to operation in conjunction" are beyond copyright's scope of protection. The Ninth Circuit similarly held in *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1522 (9th Cir. 1992) that interface features required for interoperability were functional elements unprotectable under § 102(b). With these rules in place, innovators can take advantage of and further develop robustly interoperable systems, fueling competition. Without them, a few market incumbents could introduce "control points" to create barriers for market newcomers.

> **2. Open Source Development and Open Standards Demonstrate the Need for Clear Copyright Limitations to Support and Fuel Competitive Market Alternatives.**

Open source software (OSS) development and open technological standards each provide important examples of how innovation flourishes when interoperability is strong, and of how introducing "control points" can limit competition. OSS development follows a collaborative innovation model that

maximizes interoperability between programs by using open licenses. The licenses

ensure developers can modify and redistribute the program's source code,

maximizing interoperability and follow-on development. *See* Open Source

Initiative, *The Open Source Definition*.[24]

Via this model, OSS innovation has transformed the technology industry by

injecting highly innovative programs into the software market. Open source can be

found at every level of systems, including operating systems (Linux), databases

(MySQL), web server software (Apache), Internet browsers (Firefox), word

processing programs (OpenOffice), and programming languages (Perl). OSS

makes up virtually the entire backbone of the Internet. *See, e.g.*, Tim O'Reilly, *The*

Open Source Paradigm Shift, in *Perspectives on Free and Open Source Software*

461 (J. Feller, B. Fitzgerald, S. Hissam, & K. R. Lakhani, eds., 2007). By some

estimates, OSS returns savings of $60 billion annually to software users. *See*

Richard Rothwell, *Creating Wealth With Free Software*, Free Software Magazine

(Aug. 5, 2008).[25]

Importantly, OSS represents strong competitive alternatives in the software

market. Samba,[26] an OSS project, presents users with an important alternative way

to connect Linux and Windows-based computers. *Amicus* Mozilla provides a

---

[24] *Available at* http://opensource.org/osd.
[25] *Available at*
http://www.freesoftwaremagazine.com/articles/creating_wealth_free_software.
[26] *See* Samba, http://www.samba.org.

variety of important OSS alternatives, including: the Firefox browser, Thunderbird

e-mail client, the SeaMonkey Internet application suite, and the FileZilla FTP

client.[27] Word processing software OpenOffice[28] competes with Microsoft's

Word. Open source blogging platform WordPress[29] is widely used; indeed,

Microsoft now uses WordPress in place of its own Live Spaces offering.

In order to build a truly broad set of competitive alternatives, however, open

source developers must be able to write code that interoperates with "proprietary"

software—that is, software that is not licensed under open source terms. It is

imperative that the methods of operation and other functional elements of the code

remain available for use outside of proprietary licenses. Should this change,

owners of market-dominant proprietary software would have more power to

restrict the types of open source software that can communicate with their

software. For example, OSS projects like the Firefox browser and OpenOffice rely

on interacting with proprietary computer operating systems in order to compete

with other programs. In the case of OpenOffice, this interaction directly allows

competition with Microsoft's proprietary Word product.

Technologies provided as open standards similarly fuel innovation and

competition by enhancing interoperability. For example, in retail markets,

---

[27] *See* Mozilla Products, http://www.mozilla.org/en-US/products/#tools.
[28] *See* Apache Software Foundation, Open Office, http://www.openoffice.org/
[29] *See* WordPress, http://wordpress.org/

companies historically communicated to each other via proprietary systems that

were limited in the syntax of their messaging and undecipherable to outside parties.

This added cost and complexity to placing products from new market entrants into

stores. By using the extensible markup language (XML) to design a set of

information components commonly used in business transaction documents (like

"party", "item", "address", etc.) and specifying the rules for combining them,

amicus Glushko and others created a set of documents that made business

information exchanges more efficient. They called their XML language the

Common Business Language, and today its successor, the Universal Business

Language, contains scores of standard business documents used in many business

systems all over the world.

Today, XML is an open standard that has been used to create thousands of

such domain-specific languages. *See* W3C Standards, *XML Open Status*.[30] It is the

default format for Microsoft Office, Open Office, and Apple iWork, and for critical

Web services systems such as RSS and XHTML. *See* CoverPages.org, *XML*

Applications and Initiatives.[31] Amazon has adopted XML for the Amazon

Marketplace platform, which serves thousands of merchants and is the largest

marketplace of its kind, to facilitate sales by individual merchants in the Amazon

---

[30] *Available at* http://www.w3.org/standards/techs/xml#w3c_all
[31] *Available at* http://xml.coverpages.org/xmlApplications.html.

online store. *See* Amazon Services, *Selling on Amazon Guide to XML.*[32] This

boosts competition and brings beneficial products to consumers.

In order for XML to continue to provide these benefits, however, users must

be able to rely on the rules that dictate its use. This includes both its ongoing status

as an open standard, and on the underlying rules that dictate which of its elements

are copyrightable. Preserving limits on copyrightability allows this certainty and

prevents centralizing control of XML after-the-fact—precisely what Oracle has

attempted with the Java APIs at issue in this case. If users of XML had not been

able to rely on the rules surrounding its use, it could not have enabled the broad

interoperability—and the competition and innovation—that it supports today.

Open source and reliable open standards provide important examples of the

stakes at issue for competition in this case, but these stakes apply more broadly.

Expanding copyrightability beyond its traditional bounds to elements essential for

communication between software—such as the Java API elements at issue here—

could create the type of "control points" that allow market incumbents to limit

entrance by competitors more generally, slowing innovation and harming

consumers. Regardless of their business models, developers would be limited in

their ability to connect competing products to existing networks, harming

competition.

---

[32] *Available at* https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/XML_Documentation_Intl.pdf.

3. Appropriate Limitations on Copyrightability Encourage Competition by Preventing Consumer Lock-In.

For a competitive marketplace to function, consumers must be able to switch between products without substantial detriment. As described above, clear and stable limitations on copyrightability can help keep markets competitive, ensuring that consumers can choose among products.

Consumers should also be able to transfer time and skills they invest into commercial products without then being then locked into that particular product. Consumers commonly engage in their own innovation on top of existing products they own. *See* Eric von Hippel, *Democratizing Innovation* at 4 (2006) (cataloguing instances of end-user innovation and explaining that empirical studies show that as many as 40 percent of users engage in modifying products). End-user innovation makes products more useful and valuable to their owners. It also creates broader benefits by spurring product improvement, especially where features or uses do not present justifiable investments for corporations.

But valuable user investments become sunk costs—and increase lock-in—if user-innovators cannot then transfer their innovations to another product choice. For example, such a situation could have occurred with the spreadsheet macros at issue in *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 814 (1st Cir. 1995). Macros are sequences of stored commands created by a user to automatically

instruct a computer program. Lotus 1-2-3, one of the first spreadsheet programs, allowed users to create macros through its menu command hierarchy. Lotus 1-2-3 customers implemented their own innovations on the Lotus platform by creating macros.

For those who wanted to move to Borland's competing spreadsheet, these user innovations could have been thwarted by Lotus's attempt to expand copyrightability by claiming copyright in the 1-2-3 menu command hierarchy. However, as the court noted, "Original developers are not the only people entitled to build on the methods of operation they create; anyone can." *Lotus*, 49 F.3d at 818. Because the hierarchy was correctly recognized as an uncopyrightable method of operation, *Lotus* at 815, user-innovators could avoid lock-in and choose other spreadsheet programs on which to run their macros.[33]

B. Appropriate Limits on Copyrightability Are Necessary to Ensure that Future Competitive Products and Services Can Develop.

One of the most important benefits copyright's traditional limitations provide is protecting technical innovations that are in the very early stages—or still beyond the horizon. For example, "cloud services" employ computing resources

---

[33] *Lotus* has been cited with approval by numerous circuit courts. *See Hutchins v. Zoll Medical Corp.*, 492 F.3d 1377 (Fed. Cir. 2007); *Incredible Techs., Inc. v. Virtual Techs., Inc.*, 400 F.3d 1007 (7th Cir. 2005); *Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004); *MiTek Holdings, Inc. v. Arce Engineering Co.*, 89 F.3d 1548 (11th Cir. 1996).

that are delivered as a service over the Internet, allowing both business and

individual users to expand computing power and storage capacity. Millions of

users employ cloud services to store backups, access web-based applications, and

provide services or content. Start-ups like those supported by the investor *amici*

often leverage cloud platforms' services—usually via APIs—to achieve scalability

that would not be possible using in-house systems.

Competition in the burgeoning cloud services market depends, in part, on

copyright limitations. Cloud services rely heavily on interoperability—and

specifically on APIs—to work. If the elements of the Java API at issue were held

to be copyrightable, the ensuing uncertainty would seriously undermine the active

innovation and robust competition that are current features of the cloud market. *See*

Robert McMillan, *Could an Oracle Win Against Google Blow Up the Cloud?*,

Wired (June 15, 2012).[34]

Another developing platform, this one still just over the horizon, is the so-

called "Internet of things"—the networking of devices and objects via the Internet.

See Kevin Ashton, *That 'Internet of Things' Thing*, RFID Journal (June 22,

2009).[35] Virtually any device or object—from smartphones to supply-chain tags to

automobiles to home appliances—can provide and receive information through the

Internet. *See* RFID Working Group Of The European Technology Platform On

---

[34] *Available at* http://www.wired.com/wiredenterprise/2012/05/oracle_clou/.
[35] *Available at* http://www.rfidjournal.com/article/view/4986.

Smart Systems Integration, *Internet of Things in 2020* 5-7 (2008).[36] Commentators

note that this promises numerous benefits to society, such as automated cars,

"smart" supply chains, and consumer electronics that monitor power consumption

and track and replace household items as they run low. *See Architecting the*

Internet of Things (Dieter Uckelmann, Mark Harrison and & Florian Michahelles

eds. 2011).[37] Preserving the robust interoperability required to build such a diverse

and expansive network—and the innovation and investment needed for that

building—requires exactly the kind of certainty about copyright's parameters

implicated by this case.

III. The District Court Properly Applied Established Copyrightability Limits and Correctly Separated Protectable Creative Expression from Unprotectable Elements.

The scope of copyright protection for computer programs has always been

carefully and purposefully limited. As noted in *Johnson Controls, Inc. v. Phoenix*

Control Sys., Inc., 886 F.2d 1173, 1175 (9th Cir. 1989), "[w]hether a particular

component of a program is protected by a copyright depends on whether it

qualifies as an 'expression' of an idea, rather than the idea itself." And to be

copyrightable, elements of a computer program must not fall under important

---

[36] *Available at* http://www.smart-systems-integration.org/public/documents/publications/Internet-of-Things_in_2020_EC-EPoSS_Workshop_Report_2008_v3.pdf.

[37] *Available at* http://www.cui-zy.cn/Recommended/Linux/Architecting_the_Internet_of_Things.pdf.

statutory categories, such as procedures, processes, systems, and methods of operation. 17 U.S.C. § 102(b). Reversing the district court's ruling would represent a dramatic shift in copyright law, as it would extend protection beyond current limitations that were carefully crafted to protect innovation.

A. Reversing the District Court's Ruling Would Upset a Longstanding and Carefully Constructed Balance for Copyright Protection in Computer Programs.

Legislators and courts have consistently voiced grave concern that overly broad copyright protection for computer programs would hinder innovation. Section 102(b) was the result of strong witness opinions in Senate hearings advocating for careful statutory demarcation around the scope of copyright if it were to be extended to computer programs. *See Copyright Law Revision: Hearings on S. 597 Before the Subcomm. on Patents, Trademarks, and Copyrights of the S. Comm. on the Judiciary*, 90th Cong. 1059 (1967) (hereinafter *Senate Copyright Hearings*) (statement of W. Brown Morton, Jr.). The House Report accompanying § 102(b) similarly noted the importance of properly delimiting the copyrightability of computer programs. *See* H.R. Rep. No. 94-1476 at 56–57 (1976).

Section 102(b) was directly intended to address these concerns. *See Senate Copyright Hearings* at 969–74 (testimony of Arthur A. Miller); H.R. Rep. No. 94-1476 at 56–57 ("Section 102(b) is intended, among other things, to make clear that

the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law."); *see also* Pamela Samuelson, *Why Copyright Law Excludes Systems and Processes from the Scope of Its Protection*, 85 Tex. L. Rev. 1921, 1944–55 (2007).

Courts have also addressed such concerns, by policing copyrightability limits for computer programs, including limitations that specifically support interoperability.[38] *See* 17 U.S.C. § 102(b); *Sony Computer Entm't, Inc., v. Connectix Corp.*, 203 F.3d 596, 602 (9th Cir. 2000); *Lotus*, 49 F.3d at 815; *Altai*, 982 F.2d at 707–08; *Sega*, 977 F.2d at 1522.

B.   The Elements of the Java APIs at Issue in this Case Are Uncopyrightable Methods of Operation.

Among other benefits, keeping copyrightability limitations stable preserves certainty for innovators and prevents elements essential to basic interoperability from becoming the "control points" described above in Section II.

Lotus v. Borland provides a particularly relevant application of this principle. *See* 49 F.3d at 815. In *Lotus*, the court found that the menu command hierarchy of the spreadsheet program Lotus 1-2-3 was an uncopyrightable method

---

[38] In so doing, neither Congress nor courts have forced entities to choose between patent protection and copyright protection for software programs. Instead, courts have recognized that expressive elements of software are protected while functional elements are not (although they may be patentable).

of operation because it was essential to making use of the program's functional

capabilities. *See id.* The hierarchy did not "merely explain and present Lotus 1-2-

3's functional capabilities to the user;" instead, it "serv[ed] as the method by which

the program is operated and controlled." *Id.* The court stated that "if specific words

are *essential to operating something*, then they are part of a 'method of operation'

and, as such, are unprotectable." *Id.* at 816 (emphasis added).

Methods of operation do not themselves become copyrightable because they

relate to some creative expression. *See id.* ("The 'expressive' choices of what to

name the command terms and how to arrange them do not magically change the

uncopyrightable menu command hierarchy into copyrightable subject matter.").

Nor do available alternative design choices affect whether a program feature is a

method of operation. *Id.* ("The fact that Lotus developers could have designed the

Lotus menu command hierarchy differently is immaterial to the question of

whether it is a 'method of operation.'")

The Java API elements at issue here are comparable to the menu hierarchy in

Lotus: uncopyrightable because they constitute the method of operation through

which a user's program accesses, controls, and makes use of the functional

capabilities of the Java API. A program that uses the Java API calls upon a method

in the API through the method's header, which consists solely of the names of the

method, class, and package the method is classified under: uncopyrightable features that must be operated for a program to use a function within the API.

C. **The Elements of the Java APIs at Issue in this Case Are Uncopyrightable Because They Are Dictated by Efficiency and Interoperability.**

In addition to being uncopyrightable as a method of operation, the elements of the Java API at issue are uncopyrightable as elements that are "dictated by efficiency," causing them to merge with the ideas underlying them. *See Altai*, 982 F.2d at 707. The court in *Altai* reasoned that since "efficiency is akin to deriving the most concise logical proof or formulating the most succinct mathematical computation . . . the more efficient a set of modules are, the more closely they approximate the idea or process embodied in that particular aspect of the program's structure." *Id.* at 708.

Here, the structure, sequence, and organization of the 37 Java API packages copied by Google act in exactly this way. Indeed, they were developed to maximize efficiency and facilitate software programming on the Java platform. *Oracle*, 872 F. Supp. 2d at 1000. As the district court recognized, "millions of lines of code" had been written in Java before the arrival of Google's Android operating system. *Id.* As a result, a great number of developers are well versed in designing programs on the Java platform. In order for some of these programs to correctly function and for developers to seamlessly across platforms, "Google was required

to provide the same . . . command system using the same names with the same 'taxonomy' and with the same functional specifications." *Id.*

Similarly, Google replicated what was necessary to achieve a degree of interoperability. *Id.* In ruling that the command structure of the Java API at issue is an uncopyrightable method of operation, Judge Alsup properly noted the importance of the fact that "[d]uplication [of the structure] is necessary for interoperability." *Oracle*, 872 F. Supp. 2d at 977. In the Ninth Circuit, which controls here, Section 102(b) bars from copyright protection software interfaces necessary for interoperability. In *Sega,* elements of the Sega video game interface necessary for "compatibility"—*i.e.*, interoperability—were functional aspects not copyrightable under Section 102(b). *Id.* at 1522 ("Accolade copied Sega's software solely in order to discover the functional requirements for compatibility with the Genesis console—aspects of Sega's programs that are not protected by copyright. 17 U.S.C. § 102(b)."). Likewise, in *Connectix*, functional elements that established interoperability with the Sony Playstation's operating software were unprotected. *Connectix,* 203 F.3d 596. [39]

---

[39] Although the Ninth Circuit in *Sega* and *Connectix* discussed interoperability in the fair use section of their analysis, both cases addressed copyrightability as a threshold question, not as a fair use factor. Both cases stand for the same conclusion: elements of software code essential to interoperability are uncopyrightable as a matter of law. Thus, the rulings in *Sega* and *Connectix* are, contrary to Oracle's claims, directly relevant to consideration of the copyrightability of the APIs at issue in this case.

No part of this legal framework is new. It has been well-established for over

130 years that the idea behind an original expression is uncopyrightable. *Baker v.*

Selden, 101 U.S. at 103. Section 102(b) is a key feature of the 1976 Copyright Act.

Altai provides the standard approach courts apply in determining the

copyrightability of elements of computer programs and filtering out unprotectable

procedures, processes, systems, and methods of operation. *See* Samuelson, *supra* at

1973 (stating that as of 2006 *Altai* has been followed in 49 subsequent cases).

Oracle's attempts to characterize the district court decision as using unreliable

precedent to develop a novel formulation of copyrightability limits for computer

software, Appellant's Br. at 60-6, are simply incorrect. *See also* Br. *Amici Curiae*

Intellectual Property Professors in Supp. of Def.-Cross Appellant and Affirmance.

It is the changes to well-settled limitations on copyright in computer

programs that would disrupt the existing framework and create uncertainty for

innovators and their investors alike.

CONCLUSION

A stable, properly balanced regime of copyright protections and limitations

is required for innovation and investment in the software sector to continue to

flourish. Oracle's attempt to redraw established boundaries around copyright

protection for software programs would introduce chilling uncertainty into

software development. It would call into question innovators' ability to use widely

available API elements necessary for interoperability, and would introduce further

uncertainty regarding the scope of copyright protection for computer programs in

general. The district court prevented this result by correctly applying copyright's

clear, longstanding limits on copyrightability. This Court should *affirm*.

Respectfully Submitted,

May 31, 2013   /s/ Jennifer M. Urban
                             Jennifer M. Urban
Samuelson Law, Technology and Public Policy
Clinic U.C. Berkeley School of Law
585 Simon Hall
Berkeley, CA 94720-7200
Telephone: 510-642-7338
Facsimile: (510) 643-4625
Counsel of Record

CERTIFICATE OF SERVICE & CM/ECF FILING

2013-1021, -1022

I hereby certify that I electronically filed the foregoing Brief of *Amicus*

Curiae using the Court's Case Management/Electronic Case Filing system, which

will send a "Notice of Docket Activity" to the below-listed counsel:

Counsel for Defendant-Cross-Appellant Google Inc.

Christa M. Anderson
Counsel of Record
Robert A. Van Nest
Steven A. Hirsch
Michael S. Kwun
Dan Jackson
Keker & Van Nest LLP
633 Battery Street
San Francisco, CA 94111-1809
(415) 391-5400
canderson@kvn.com
rvannest@kvn.com
shirsch@kvn.com
mkwun@kvn.com
djackson@kvn.com

Renny Hwang
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
(650) 253-2551
nnyhwang@google.com

Daryl L. Joseffer
Bruce W. Baber
King & Spalding LLP
1700 Pennsylvania Avenue, N.W.
Washington, D.C. 20006
(202) 737-0500
bbaber@kslaw.com

Ian C. Ballon
Heather Meeker
Greenberg Traurig, LLP
1900 University Avenue, 5th Floor
East Palo Alto, CA 94303
(650) 328-8500
ballon@gtlaw.com
meekerh@gtlaw.com

Counsel for Appellant Oracle America, Inc.

E. Joshua Rosenkranz
Counsel of Record
drew D. Silverman
Orrick, Herrington & Sutcliffe LLP
51 West 52nd Street
New York, NY 10019s
212-506-5380
jrosenkranz@orrick.com
asilverman@orrick.com

Annette Louise Hurst
Orrick, Herrington & Sutcliffe LLP
405 Howard Street
San Francisco, CA 94105
415-773-4585
ahurst@orrick.com
Elizabeth Cincotta McBride
Gabriel Morgan Ramsey
Orrick, Herrington & Sutcliffe LLP
1000 Marsh Road
Menlo Park, CA 94025
650-614-7377
emcbride@orrick.com
gramsey@orrick.com

Mark S. Davies
Orrick, Herrington & Sutcliffe LLP
Columbia Center
1152 15th Street, N.W.
Washington, DC 20005
202-339-8631
mark.davies@orrick.com

Dale M. Cendali
Joshua L. Simmons
Kirkland & Ellis LLP
601 Lexington Avenue
Citigroup Center
New York, NY 10022
212-446-4800
dale.cendali@kirkland.com
joshua.simmons@kirkland.com

Dorian Estelle Daley
Deborah Kay Miller
Matthew Sarboraria
Andrew C. Temkin
Oracle America, Inc.
Oracle Legal Department
500 Oracle Parkway
Redwood Shores, CA 94065
650-506-5200
dorian.daley@oracle.com
deborah.miller@oracle.com
matthew.sarboraria@oracle.com
andrew.temkin@oracle.com

Susan M. Davies
Kirkland & Ellis LLP
655 15th Street, N.W.
Washington, DC 20005
202-879-5000
susan.davies@kirkland.com

Kenneth Alexander Kuwayti
Morrison & Foerster LLP
755 Page Mill Road
Palo Alto, CA 94304-1018
650-813-5600
KKuwayti@mofo.com

Michael Allen Jacobs
Morrison & Foerster LLP
Firm: 415-268-7178
425 Market Street
San Francisco, CA 94105
415-268-7455
mjacobs@mofo.com

Sean Fernandes
Kirkland & Ellis LLP
3330 Hillview Ave.
Palo Alto, CA 94304
650-859-7014
sean.fernandes@kirkland.com

Diana M. Torres
Kirkland & Ellis LLP
333 S. Hope Street
Los Angeles, CA 90071
213-680-8400
diana.torres@kirkland.com

Counsel for Amici Curiae

Jared Bobrow
Aaron Y. Huang
Weil, Gotshal & Manges LLP
201 Redwood Shores Parkway
Redwood Shores, CA 94065
650-802-3000
jared.bobrow@weil.com
aaron.huang@weil.com
Counsel for Amici Curiae
Eugene H. Spafford, et al.

Matthew S. Hellman
Paul March Smith,
Jenner & Block LLP
1099 New York Avenue, NW,
Suite 900
Washington, DC 20001
202-639-6861
mhellman@jenner.com
mith@jenner.com
Counsel for Amicus Curiae
BSA - The Software Alliance

William A. Rudy
Lathrop & Gage, LC
2345 Grand Boulevard
Kansas City, MO 64108
816-292-2000
wrudy@lathropgage.com
Counsel for Amici Curiae
Picture Archive Council of America,
Inc., et al.

Brianna E. Dahlberg
Carole E. Handler
Lathrop & Gage LLP
1888 Century Park East, Suite 1000
Los Angeles, CA 90067
310-789-4600
chandler@lathropgage.com
bdahlberg@lathropgage.com
Counsel for Amici Curiae
Picture Archive Council of America,
Inc., et al.

Gregory G. Garre
Lori Alvino McGill
Latham & Watkins LLP
Suite 1000
555 Eleventh Street, NW
Washington, DC 20004
202-637-2200
gregory.garre@lw.com
lori.alvino.mcgill@lw.com
Counsel for Amicus Curiae
Microsoft Corporation

Marcia Beth Paul
Deborah A. Adler
Lacy H. Koonce
Davis Wright Tremaine, LLP
1633 Broadway, 27th Floor
New York, NY 10019
212-489-8230
marciapaul@dwt.com
rahadler@dwt.com
koonce@dwt.com
Counsel for Amicus Curiae
Ralph Oman

Douglas B. Luftman
NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089
408-822-6521
Douglas.luftman@netapp.com
Counsel for Amicus Curiae
NetApp, Inc.

Paul T. Dacier
Krishnendu Gupta
EMC Corporation
176 South Street
Hopkington, MA 01748
paul.t.dacier@emc.com
krish.gupta@emc.com
Counsel for Amicus Curiae
EMC Corporation

Steven Thomas Cottreau
Clifford Chance Rogers & Wells LLP
The William Rogers Building
2001 K Street, N.W.
Washington, DC 20005-1001
202-912-5000
steve.cottreau@cliffordchance.com
Counsel for Amici Curiae
Scott McNealy, et al.

Upon acceptance by the Court of the e-filed document, six paper copies

will be filed with the Court, via Federal Express, within the time provided in the

Court's rules.

June 7, 2013   /s/ Stephen Moore
                                                         Stephen Moore

CERTIFICATE OF COMPLIANCE

As counsel for *Amici Curiae* Software Innovators, Start-ups, and Investors, I hereby certify the brief contained herein complies with the typeface requirements of Fed. R. App. P. 29(c) and 32(a)(5) because it has a proportionally spaced 14-point typeface.

I further certify that the body of this brief complies with the type-volume limitations of Fed. R. of App. P. 29(d) because it contains 6,667 words, based on the Word Count feature of Microsoft Word 2011. Pursuant to Federal Rule of Appellate Procedure 32(a)(7)(B)(iii) and Federal Circuit Rule 32(b), this word count does not include the words contained in the Certificate of Interest, Table of Contents, and Table of Authorities.

May 31, 2013   /s/ Jennifer M. Urban
                                     Jennifer M. Urban
                                     Counsel of Record
                                     SAMUELSON LAW, TECHNOLOGY AND PUBLIC
                                     POLICY CLINIC
                                     U.C. Berkeley School of Law
                                     585 Simon Hall
                                     Berkeley, CA 94720-7200
                                     Telephone: (510) 642-7338
                                     Facsimile: (510) 643-4625