No. 18-956

IN THE

Supreme Court of the United States

GOOGLE LLC,

PETITIONER,

v.

ORACLE AMERICA, INC.,

RESPONDENT

On Writ of Certiorari to the United States Court of Appeals for the Federal Circuit

BRIEF OF *AMICI CURIAE* SMALL, MEDIUM, AND OPEN SOURCE TECHNOLOGY ORGANIZATIONS IN SUPPORT OF PETITIONER

JASON M. SCHULTZ
(COUNSEL OF RECORD)
CHRISTOPHER J. MORTEN
NYU TECHNOLOGY LAW AND POLICY CLINIC
NYU SCHOOL OF LAW
245 SULLIVAN STREET, 609
NEW YORK, NY 10012
TELEPHONE: (212) 992-7365
JASON.SCHULTZ@LAW.NYU.EDU

Counsel for Amici Curiae

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

INTERESTS OF *AMICI CURIAE*[1]

Amici are small, medium, and open source technology organizations.

Mozilla Corporation has been a pioneer and advocate for the web for more than a decade. Mozilla creates and promotes open standards that enable innovation and advance the web as a platform for all. Today, hundreds of millions of people worldwide use Mozilla Firefox to discover and experience the web on computers, tablets, and mobile phones.

A Medium Corporation (Medium) provides an online publishing platform where people can read, write, and discuss the ideas of the day. Medium's ecosystem connects users with thoughtful, long-form writing by leaders, thinkers, entrepreneurs, artists, and journalists. Over 100 million people read on Medium each month.

At Cloudera, we believe that data can make what is impossible today, possible tomorrow. We empower people to transform complex data into clear and actionable insights. Cloudera delivers an enterprise data cloud for any data, anywhere, from the Edge to AI. Powered by the relentless innovation of the open source community, Cloudera advances

[1] No counsel for a party authored this brief in whole or in part, and no such counsel or party made a monetary contribution intended to fund the preparation or submission of this brief. No person other than the *amici curiae* or their counsel made a monetary contribution to its preparation or submission. The parties have consented to the filing of this brief.

digital transformation for the world's largest enterprises.

Creative Commons is a nonprofit providing the standard legal and technical infrastructure on the web that makes making sharing and innovation possible, including the use of APIs for accessing and sharing works that individual creators have voluntarily declared they seek to openly share under a Creative Commons license. APIs provide a common system of communication in our everyday connected world, without which sharing is disabled and friction dominates, and creators' intentions are thwarted.

Shopify Inc. is a leading global commerce company, providing Internet-based software tools to help start, grow, and manage a retail business of any size. Shopify strongly believes in the value of an open Internet, and more than one million merchants rely on Shopify's tools to integrate with thousands of third party sales channels, marketing platforms, payment gateways, and other online applications.

Etsy, Inc., and the 2.6 million creative entrepreneurs who actively sell on Etsy, rely on open standards to help make Etsy's marketplace flourish. Etsy's marketplace connects millions of buyers to sellers from nearly every country in the world for unique, special products.

Reddit provides an online network of communities where over 430 million people every month find experiences built around their interests, hobbies and passions. In operating its services, Reddit uses APIs developed by others in countless ways. Reddit itself also develops APIs through which other services interact with Reddit.

The Open Source Initiative (OSI) is a nonprofit organization founded in 1998 in order to promote the benefits of open source software through both education and advocacy. OSI also acts as a standards body by maintaining the Open Source Definition, an industry standard that encourages trust among developers, users, corporations, and governments, and that facilitates open source cooperation. The maintenance of this standard allows for the flourishing of alternatives to proprietary software that expand choice in the marketplace, spurring competition and promoting progress of computer arts and sciences.

Mapbox is a growing startup founded in Washington, D.C., with more than 500 million users interacting with its technology each month. Despite offering products that compete with Google Maps, Mapbox's interests in this case concern the bigger picture. Balance and predictability in copyright law are vital to innovation as a whole in the software industry. As a provider of online services, Mapbox is intimately familiar with APIs, providing many such interfaces to its customers. The possibility of copyright protection did not motivate Mapbox to make these interfaces; ease of use for customers did.

Patreon is a membership platform that makes it easy for creators to get paid by their fans. Patreon has sent over $1 billion to creators since its founding, which is made possible because of the many API-based integrations with its partners to allow creators to offer membership across the internet.

Wikimedia Foundation is a non-profit organization based in San Francisco, California, which operates twelve free-knowledge projects on the Internet, including Wikipedia. Wikimedia's mission is to develop and maintain factual and educational content created and moderated by volunteer contributors, and to provide this content to people around the world free of charge. Additionally, the Foundation writes free and open source software to enable people worldwide to implement wiki-style information exchanges for their own usage. The MediaWiki software that the Foundation develops, including APIs, has been implemented by corporations, educational institutions, and government agencies to record and share information, and the Foundation encourages such use as in line with its mission to share knowledge.

Software Freedom Conservancy (Conservancy) is a charity dedicated to helping people take control of their computing experience by supporting, creating and defending free and open-source software developed by volunteer communities and licensed for the benefit of all. Conservancy is the nonprofit home for over 40 free and open-source projects and initiatives such as Git, Inkscape, Busybox, Homebrew, Samba, QEMU and Selenium, which include thousands of volunteer contributors. Conservancy's communities maintain some of the most fundamental utilities in computing today and introduce innovations that shape software for the future.

SUMMARY OF ARGUMENT

Competition and innovation are at the heart of a healthy internet and the field of software development that fuels it. For decades, software engineers have relied heavily on reimplementation,[2] including reuse of functional protocols such as the software interfaces in this case, to create competing alternatives to incumbent industry players and develop new markets without fear of copyright infringement. In accord with this Court's ruling in *Baker v. Selden*, 101 U.S. 99, 105 (1879), and the plain language of 17 U.S.C. § 102(b), the software industry has flourished utilizing this approach to make internet and software solutions more accessible, affordable, diverse, and robust.

By reversing this rule in the context of Application Programming Interface (API) packages,[3] the Federal Circuit upended decades of industry practice and the well-established expectations of developers, investors, and consumers. Reimplementation is a standard practice among software developers—from those wishing to create entirely new platforms to those wishing to make their platforms compatible with other developers' software. The court's decisions below heedlessly

[2] Reimplementation, in the software industry, is the "process of writing new software to perform certain functions of a legacy product . . . . Through reimplementation, the new entrant creates its own computer code to perform the functions, but reuses the limited number of instructions that are required to create the interface already known by the users." Pet'r's Br. 7.
[3] Consistent with the Federal Circuit's opinions, "API packages" in this brief should be understood as "Java SE Libraries," as that term is used in Petitioner's brief. *See* Pet'r's Br. 8 n.5.

unraveled this reasonably predictive rule and the set of reliable norms it informs, which allow software coders to understand what is appropriate to carry over from one project to another and what is not. This is especially true for small and medium technology enterprises (SMEs) and open source software developers, who are often highly sensitive to new litigation risks.

Amici urge the Court to reverse the Federal Circuit's decisions and correct this misreading of copyright law. Amici raise two fundamental concerns with the Federal Circuit's reasoning. First, the Federal Circuit's dramatic expansion of copyright protection to include functional elements of API packages, which Amici believe are not copyrightable under U.S. law, stifles innovation and competition by privileging powerful incumbents, creating artificial barriers to entry for new players, and deterring new software development. Second, the court's rejection of the fair use doctrine stands to undermine not only reimplementation of API packages, but also other valuable software engineering practices related to reverse engineering, interoperability, and creation of competing platforms, as well as innovations in data analysis, search engines, and many other groundbreaking advancements. In doing so, the Federal Circuit has opened the door to relitigating many status quo software engineering practices—practices that SMEs and open source projects depend on every day to produce new platforms, programs, features, and interfaces. *See, e.g.*, *Sony Comput. Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596, 606-07 (9th Cir. 2000); *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1522-23 (9th Cir. 1992).

For these reasons, we urge the Court to reverse the Federal Circuit's decision below and rule in favor of Google.

ARGUMENT

I. Introduction

APIs facilitate countless functions and innovations in the software world. From helping the software running your phone to maintaining medical equipment to supporting every electronic connection your computer makes to another device, it would be impossible to list them all. At issue in this case is a particular set of API packages for a mobile operating system, but the implications of the Federal Circuit's rulings are much larger and have the potential to completely transfigure software production, competition, and innovation, especially on the internet.

To help illuminate these concerns, we encourage the Court to understand software interfaces (which are a type of API) as similar to the electronic checkout forms you see when shopping online. When you buy a product on an e-commerce website, you are typically asked to enter information related to method of payment and shipping. While every e-commerce site has a slightly different style, payment screens almost without exception ask you to fill out a nearly identical structured form: name, address, credit card or bank information, billing address, shipping address, etc. E-commerce sites will display these fields in various shapes, sizes, fonts, and colors, but the structure, sequence and organization (SSO) of the information have become conventions. While a shopping site could attempt to come up with a totally new format for requesting billing and shipping data, common sense, technological standardization, and economic efficiency have driven the industry to adopt an almost ubiquitous SSO that every user expects, understands, and completes with ease.

Now imagine that the copying of this SSO from one site to another is deemed to constitute copyright infringement. This would force every website with a payment page to either pay licensing fees to the original inventor of the standard structure or invent a unique SSO for payment and shipping forms, requiring users to enter information in repeatedly unfamiliar formats with unintuitive naming conventions. Instead of entering "First Name, Last Name," for example, users might be required to enter "Name as it appears on most recent 1040 tax form for U.S. taxpayers" or "Name that comes after your first name and middle name(s)." For every new site, the problem would repeat. This would impose enormous, unwarranted costs on both creators and consumers, inefficiently redirecting the time and energy of software developers while also wasting customers' time navigating new and potentially confusing payment and shipping forms for every online store they use. Such costs could also deter customers from trying new platforms and reinforce the market power of dominant vendors, not because those vendors have better products or prices, but because they happen to have been the first to publish the SSO of the dominant payment and shipping forms. Software bugs and customer errors would both become more frequent. Similar problems would arise for nonprofit online donations pages and discussion forums, which likewise rely on the ability to freely copy the SSO of electronic interfaces to maximize public accessibility and engagement. To hold that the SSOs of each of these forms and interfaces are copyrightable and thus require licensing if not litigation would neither be a victory for innovation, creativity, efficiency, or competition, nor a result that copyright law was designed to achieve. Instead, so holding would erase many of the benefits that online technologies and marketplaces provide to society, from low barriers for new entrants to low-stress, intuitive opportunities for users.

The above concerns apply equally to SSOs for API packages. Much as developers and users of e-commerce and other payments websites have come to expect and depend on standardized SSOs for checkout forms, both developers of operating systems and developers of applications for those operating systems expect and depend on standardized SSOs for API packages when programming. For application developers, especially open source projects and SMEs, standardized SSOs are an economic necessity, since they often lack the resources to modify or adapt their applications to every bespoke platform, especially when each platform might have hundreds or even thousands of relevant API packages. When SSOs for API packages are consistent across operating systems or other platforms, application developers are able to quickly and efficiently improve or adapt original products to new marketplaces. This compatibility means software coders can develop apps for one platform knowing they will also run consistently and predictably on other platforms.

More platforms can offer a wider range of applications, providing consumers with new choices and more competition.

The Federal Circuit's rulings threaten these economic and societal benefits. If the Federal Circuit's decisions stand, developers of new operating systems or platforms will either be at the mercy of dominant players' licensing practices and prices or be forced to adopt ill-suited alternatives that will put them at an immediate disadvantage in attracting third-party applications and users they need to compete effectively. Software will likely become less diverse, more expensive, less compatible, and more error-prone. It was for these reasons that this Court decided to deny copyright to the ledger system in *Baker*, and this reasoning has (until now) led to consistent decisions across the circuit courts that have ruled on copyrightability and fair use for software. Amici urge this Court to uphold the *Baker* line of precedents, reverse the Federal Circuit's rulings, and reaffirm that copyright does not stand in the way of software developers reusing SSOs for API packages in socially, technologically, and economically beneficial ways.

II. The Court Should Reverse the Decisions Below To Prevent Chilling of Innovation and Competition in the Software Field.

A. Successful software development requires platform SSO compatibility.

Amicus Mozilla is home to a community spanning thousands of developers who write code that interacts with APIs on a daily basis. The ethos of the industry as a whole is exemplified in projects like Mozilla's Firefox browser, whose open source code receives contributions from thousands of developers both inside and outside Mozilla every day.

Many open source software projects are similar but function without corporate support. For example, amicus Software Freedom Conservancy represents over 40 such free and open source projects. These projects are loose communities of contributors, project managers, and other organizers who volunteer their time, effort, and creativity to improve the technology that powers much of the online and electronic ecosystems we use every day.[4]

In order to successfully attract new users and third-party app developers, the vast majority of software and internet technologies must achieve compatibility with current or legacy systems. This means that software developers working across the industry as a whole are constantly iterating and building on each other's code. Basic website development works this way. For a webpage to be considered successful and professional, it needs to appear correctly on every single browser and look the same on each. If each browser required different sets of instructions to display colored text or tables, for example, web developers would need to learn entirely new display instructions and code entirely new pages for each browser in the market. And new browsers would have a hard time entering the market because no websites would have pages coded for their display instructions.

[4] *See* Steven Weber, *The Success of Open Source* (2006).

Another, more specific illustration may be helpful here: Mozilla has reimplemented Google's "Extensions" API from the Chrome web browser in Mozilla's Firefox browser.[5] Mozilla's choice to support the Extensions API through reuse of the SSO allows other developers to build a single extension and then deploy it not just in Firefox but in Google's Chrome browser, Microsoft's Edge browser, and the Opera browser, with only a few small modifications.[6] This increases the number of potential extensions available to users of all browsers, allowing them to easily enhance and add new functionality to their chosen browser, or, if they wish, switch browsers without high transaction costs. Mozilla's reimplementation of the Extensions API is a classic example of reliance on established software engineering norms for a result that is not only what software engineers expect, but that makes sense for the success of the field as a whole, with obvious benefits to competition and innovation in the market for web browsing.

Yet the Federal Circuit's decisions threaten to disrupt the industry norm of reimplementation. In ruling both that the SSO of Oracle's Java SE Libraries was copyrightable and that the fair use

[5] *Browser Extensions*, MDN Web Docs, *available at* https://developer.mozilla.org/en-US/Add-ons/WebExtensions (last visited Jan. 10, 2020).

[6] *See, e.g.*, Microsoft Edge (EdgeHTML) extensions, Microsoft Docs, *available at* https://docs.microsoft.com/en-us/microsoft-edge/extensions (last visited Jan. 10, 2020); *Porting an Extension from Chrome to Microsoft Edge*, Microsoft Docs, https://docs.microsoft.com/en-us/microsoft-edge/extensions/guides/porting-chrome-extensions (last visited Jan. 10, 2020).

defense was unavailing to petitioner, the Federal Circuit issued an edict that is nonsensical and counterintuitive for most software developer communities. Expanding copyright to include SSOs of API packages does not provide incentives to create *useful* software that has economic and social benefits. Instead, it erects new barriers to such creation and forces the creation of fragmented, complicated, and cumbersome SSO ecosystems that are completely unnecessary and unproductive from a software engineering perspective. Much like forcing every e-commerce website to create new idiosyncratic shipping and payment forms, copyright for SSOs of API packages results in confusion, wasted effort, and significant legal uncertainty in the software field, especially for open source, nonprofit, and SME developers.

B. Clear copyright rules contribute meaningfully to open source and SME software development.

While this case pits two technology giants against each other, Amici urge the Court to look beyond them and consider the importance of the issues presented to smaller players in the software industry—individual developers, startups, nonprofit, SMEs, and open source software developers, among others. In particular, we urge the Court to consider the benefits that clear copyright rules provide for software development.

Clear copyright rules are critical to the survival of small technology companies and open source projects. These efforts often start with quintessential "garage" inventors—a few individual coders huddled together in a small office or home or remotely, working to code as fast as they can to launch a new idea, product, or service into the world before their funds run out. To do this, they need copyright rules to be relatively clear. For example, software engineers generally understand they *cannot* copy someone else's application source code unless they have permission, for example, via a relevant "open source" license. Likewise, software engineers generally understand that they *may* reuse API package SSOs without a license without running afoul of copyright law, which does not apply to these functional connectors. This practice is commonplace, enabling application developers to offer their apps to consumers on a range of existing platforms, operating systems, or browsers quickly and efficiently. *See, e.g.*, Pet'r's Br. 26-27 ("[T]he decades-long understanding in the software industry has been that software functions may be freely reimplemented—and that such reimplementation 'unleashed the personal computer revolution.'"). Small innovators can "plug-and-play" their applications across all technology ecosystems without having to rewrite the mechanisms for their applications to communicate with dozens or hundreds of alternative APIs to produce identical functionality.

Under the Federal Circuit's approach, app developers will have to shoulder these additional engineering, financial, and legal costs when extending their programs to a new platform or operating system. For SMEs, open source developers and other small innovators in particular, such additional burdens may be difficult to sustain, with the effect of reducing competition and innovation in the software field, not enhancing them.

Oracle dismisses these concerns by suggesting that platform or operating system developers simply create new SSOs instead of using the SSOs for the Java SE Libraries. But, as highlighted in our payments analogy above, creation of new SSOs is a problematic workaround, an ersatz “innovation” that the marketplace neither needs nor desires. Furthermore, forcing application developers to rewrite their code for hundreds of new APIs in order to make it available on a new platform is not only burdensome and expensive, but risky, as it may create new errors or incompatibilities that will require extensive quality assurance and maintenance.

By undermining the relative clarity of copyright rules for APIs, the Federal Circuit also created new litigation risks that will disproportionately impact smaller innovators who—unlike Google and Oracle—typically have fewer resources to defend themselves. A simple cease-and-desist letter from a large software company could be enough to shut down new products or services from SMEs and open source developers, who often lack large legal teams and extensive financial reserves.

On a broader level, the Federal Circuit’s rulings threaten competition across the entire software industry. The Court’s dramatic expansion of copyright doctrine to annex the functional aspects of API packages will stifle innovation and competition by privileging powerful incumbents and creating a lock-out effect for new products. *See, e.g.,* Pet’r’s Br.

27 ("A ruling by this Court that copyright prohibits . . . reimplementation would allow the authors of older software to hold their users hostage."). This would lead to an overall decrease in choice, both for innovators and consumers. Amicus Mozilla has long advocated for technical interoperability as essential to preserving consumer choice and economic competition on many fronts.[7] Because the Federal Circuit's rulings have the potential to reinforce the dominance of industry giants by increasing their proprietary leverage over small developers and other new entrants, this Court should reverse the Federal Circuit's decisions.

[7] As Amicus Mozilla argued in a recent blog post accompanying its comment to the Federal Trade Commission on the topic of competition in the internet sector, "[i]f the future of the internet stays grounded in standards and built out through an ecosystem of transparent third-party accessible APIs, we can preserve the digital platform economy as a springboard for our collective social and economic welfare, rather than watching it evolve into an oligarchy of gatekeepers over our data." Chris Riley, *Mozilla Files FTC Comments Calling for Interoperability to Promote Competition*, Mozilla: Open Pol'y & Advoc. (Aug. 21, 2018), *available at* https://blog.mozilla.org/netpolicy/2018/08/21/mozilla-files-ftc-comments-calling-for-interoperability-to-promote-competition/ (last visited January 12, 2020); *see also* Letter from Chris Riley, Dir., Pub. Policy, Mozilla Corp., to Office of the Sec'y, Fed. Trade Comm'n (Aug. 20, 2018), *available at* https://blog.mozilla.org/netpolicy/files/2018/08/Mozilla-FTC-filing-8-20-2018.pdf (last visited January 12, 2020).

III. Reversing the Federal Circuit's Decision Would Preserve Bedrock Copyright Precedents That Software Engineers Have Relied upon for Decades.

While ostensibly limited to the legal status of the functional elements of the Java SE Libraries, the Federal Circuit's decisions on copyrightability and fair use conflict with several of the bedrock copyright precedents that software engineers throughout the entire industry rely upon every day. First and foremost, the Federal Circuit's decisions conflict with this Court's holding in *Baker v. Selden*, 101 U.S. at 105, that functional elements of a work are not copyrightable. This principle was codified at 17 U.S.C. § 102(b) and has been reaffirmed in the software context by numerous courts. *See, e.g.*, *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995), *aff'd by an equally divided court*, 516 U.S. 233, 233 (1996); *Comput. Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 707-08 (2d Cir. 1992). They also blur numerous software fair use holdings that rely on both the unprotectability of functional aspects of software and the invocation of the fair use doctrine to ensure software innovation and compatibility. *See Connectix*, 203 F.3d at 602, 606-07 (finding repeated verbatim copying of operating system code fair use for the purpose of creating compatible software); *Sega*, 977 F.2d at 1522-23 (same). At a minimum, the Federal Circuit's decisions are in tension with these holdings and could discourage reliance on them, which would, in turn, chill the innovation enabled by these longstanding holdings.

A. Open source and SME software developers benefit from this Court's rule that functional aspects of copyrighted works, including SSOs of API packages, are not protected by copyright law.

The principle this Court announced in *Baker v. Selden*—that functional aspects of copyrighted works are not protected—helped to create the modern software industry and should continue to control. 101 U.S. at 105. The Federal Circuit misreads this principle to interpret 17 U.S.C. § 102(b) as focused on the outcome of a software program instead of how it functions.

The Federal Circuit's decision on copyrightability appears to read an exception into section 102(b)'s prohibition on copyrights on "method[s] of operation," holding that "expression embodied in a method of operation" is copyrightable and then concluding that the SSO of the Java SE Libraries falls under this exception. *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1356-57 (Fed. Cir. 2014) (*Oracle I*). Going beyond SSOs for API packages, this logic could apply copyright to almost every single technical aspect of software engineering, rendering section 102(b) more or less obsolete for software. The question of how to determine which aspects fall under the Federal Circuit's exception and which do not was left entirely open by the opinions below, jeopardizing both the boundaries on copyrightable subject matter and the relative clarity that *Baker* and section 102(b) were meant to provide. Reversing the Federal Circuit and reaffirming that functionality is the guiding principle for denying copyright protection would not only align the case at hand with *Baker*, as well as *Lotus, Altai, Sega,* and *Connectix*, but would also embrace and support decades-long industry practices that have fostered one of the most successful and innovative industries in the world.

In *Baker*, this Court held that a system of double-entry book-keeping was not copyrightable (though the text describing the system of book-keeping was) because its "object . . . [was] use." 101 U.S. at 105. Thus, *Baker* set forth the foundational principle that elements of works that serve functional purposes are uncopyrightable ideas, distinct from copyrightable expression. *See id.* at 103. ("The copyright of a work on mathematical science cannot give to the author an exclusive right to the methods of operation which he propounds, or to the diagrams which he employs to explain them, . . . to prevent an engineer from using them."). By adopting section 102(b), Congress codified this principle and expressly contemplated its application in the software engineering context. H.R. Rep. No. 94-1476, at 57 (1976) ("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"); S. Rep. No. 94-473, at 54 (1975) (same). The Federal Circuit even acknowledged that the SSO of Oracle's Java SE Libraries is functional. *See Oracle I*, 750 F.3d at 1367 (observing that "Oracle does not—and concedes that it cannot—claim copyright in the idea of organizing functions of a computer program or in the 'package-class-method' organizational structure in the abstract"). The district court aptly described the SSO of the Java SE Libraries as "a command structure, a system or method of operation—a long hierarchy . . . to carry out pre-assigned functions." *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 999-1000 (N.D. Cal. 2012). As such, the SSO of the Java SE Libraries at issue simply should not be entitled to copyright protection under *Baker*.

The Court can apply the *Baker* principle to the SSOs of the Java SE Libraries by recognizing that they are uncopyrightable "method[s] of operation" under section 102(b), without, as the Federal Circuit erroneously suggests, depriving of copyright protection *per se* non-functional expression embodied in the Java SE Libraries. The First Circuit's decision in *Lotus* offers useful guidance here. *Lotus* faithfully applied *Baker* and section 102(b) when it held that the menu command hierarchy of Lotus's spreadsheet program was uncopyrightable because it provided the functional means by which users operated Lotus' program—a "method of operation." *Lotus*, 49 F.3d at 815, 817.

Lotus makes clear that only those elements of computer programs necessary for their use are uncopyrightable as a method of operation, while expressive elements not required for users to operate the program are eligible for copyright protection. *See id.* at 816 (recognizing the copyright eligibility of Lotus's "underlying computer code, because while code is necessary for the program to work, its precise formulation is not . . . [necessary for] users to operate its programs in substantially the same way"). While elements of the Java SE Libraries may be copyright eligible, their SSOs are similar to the menu hierarchies in *Lotus* and thus functionally useful in the same way. The Federal Circuit nonetheless declined to follow this principle. *See Oracle I*, 750 F.3d at 1366. This Court should apply *Baker* and the reasoning of *Lotus* to find the SSO of the Java SE Libraries unprotectable under copyright law. *See also Altai*, 982 F.2d at 707-08 (holding that all "elements dictated by efficiency" in software are uncopyrightable).

B. Even if this Court does not reverse the copyrightability ruling, it should nonetheless reverse the Federal Circuit's rejection of Google's fair use defense.

1. This Court should reverse the Federal Circuit's overly-narrow construction of transformative use, which inappropriately excludes from fair use "new opportunities" to reuse API package SSOs.

As even the Federal Circuit recognized, a secondary use of computer code is more likely to be considered fair when it "changes" the underlying copyrighted work or uses it "in a different context" so that the work is "transformed into a new creation." *Oracle I*, 750 F.3d at 1374 (quoting *Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146, 1165 (9th Cir. 2007)); *see also Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 579 (1994) ("[T]he more transformative the new work, the less will be the significance of other factors . . . that may weigh against a finding of fair use."). However, the court then misinterpreted this rule to limit its application strictly to situations where new *code* emerges from the secondary use. *Oracle I*, 750 F.3d at 1376. Such a limited and narrow ruling fails to capture the full range of transformative or "new" uses that occur in software, especially through engineering processes such as reimplementation, or when an API package SSO is reused to perform a novel *function* or expand that functionality onto a new *platform*. *See Connectix*, 203 F.3d at 606-07 (finding that simply introducing existing computer code into a new context can be transformative within the practice of software engineering); *Sega*, 977 F.2d at 1522; *see also Authors Guild v. Google, Inc.*, 804 F.3d 202, 206 (2d Cir. 2015) (finding wholesale copying of digital books for the purpose of improving search engines to be transformative); *A.V. ex rel. Vanderhye v. iParadigms, LLC*, 562 F.3d 630, 640 (4th Cir. 2009) (finding wholesale copying of student essays for the purpose of improving plagiarism detection software to be transformative); *Perfect 10,* 508 F.3d at 1165 (finding wholesale copying of millions of photographs for the purpose of improving search engines to be highly transformative).

Reimplementation of APIs to perform novel functions and expand existing functionality to new platforms are major forms of innovation in the software industry that would be imperiled if the Federal Circuit's rulings are left standing. A programmer can reuse the functional elements of code in new ways to create a new work that radically departs from the original work while retaining its basic SSO in order to ensure the old and new work remain compatible with each other. This kind of transformative departure "add[s] something new, with a further purpose or different character" to the program's SSO. *Campbell*, 510 U.S. at 579. Developers frequently innovate by introducing existing ideas into new, creative contexts, or expanding the user and developer bases for languages and functional systems.

In *Oracle America, Inc. v. Google LLC* (*Oracle II*), the Federal Circuit further unsettled the bedrock fair use holdings in *Sega* and *Connectix.* 886 F.3d 1179 (Fed. Cir. 2018) (*Oracle II*). Citing *Connectix*, *Oracle II* held that copying the SSO of the Java SE Libraries for purposes of software compatibility was not even "modestly transformative." *Oracle II*, 886 F.3d at 1200 (citing *Connectix*, 203 F.3d at 606-07). Yet *Connectix* had held that copying to create software compatibility had a transformative purpose and was protected as fair use. *Connectix*, 203 F.3d at 606-07. The defendant in that case built a new tool that rendered games written for the Sony PlayStation video game console compatible with personal computers. *See id.* at 606. The *Connectix* court wrote, "[t]his innovation affords opportunities for game play in new environments. . . . More important, the [compatibility tool] itself is a wholly new product, notwithstanding the similarity of uses and functions." *Id.* at 606; *see also Sega*, 977 F.2d at 1522 (holding that Accolade's "ultimate purpose" in copying was to create compatibility between its video games and Sega's video game console and recognizing Accolade's use as fair use). Nothing in the Federal Circuit's opinion meaningfully clarifies or reconciles the difference between its construction of *Connectix* and the *Connectix* court's holding that creating software compatibility is a transformative purpose.

2. This Court should reaffirm its analysis of the third fair use factor in *Campbell*, where it held that amount and substantiality of the original work taken need only be "reasonable" in light of the purpose, not "necessary," as the Federal Circuit erroneously held.

Finally, in addressing the third fair use factor, the Federal Circuit committed legal error in requiring secondary users of API package SSOs to show their use was "necessary" in light of their purpose, instead of merely "reasonable," as this Court held to be the test in *Campbell*. *Compare Oracle II*, 886 F.3d at 1205-06, *with Campbell*, 510 U.S. at 586 ("The third factor asks whether 'the amount and substantiality of the portion used in relation to the copyrighted work as a whole' . . . are reasonable in relation to the purpose of the copying." (citation omitted) (quoting 17 U.S.C. § 107(3))). This distinction is critical for the viability of software development. Copying practices that are reasonable, such as reuse of API package SSOs, encourage industry norms and best practices, as described above. *See supra* § II. On the other hand, requiring software engineers to prove absolute necessity before they may copy even the most basic functional SSOs would impose huge additional costs, inefficiencies, and barriers to entry, especially for small and new entrants. *Campbell*'s "reasonable" test has been in place for almost 25 years and has served the software development community well. This Court should not allow the Federal Circuit to narrow this longstanding rule arbitrarily.

CONCLUSION

For the foregoing reasons, Amici ask this Court to reverse the Federal Circuit's decisions on copyrightability and fair use.

Respectfully submitted,

JASON M. SCHULTZ
Counsel of Record
CHRISTOPHER J. MORTEN
NYU Technology Law and Policy Clinic
NYU School of Law
245 Sullivan Street, 609
New York, NY 10012
Telephone: (212) 992-7365
Jason.schultz@law.nyu.edu

Counsel for Amici Curiae