

No. 18-956

In the Supreme Court of the United States

GOOGLE LLC,

Petitioner,

vs.

ORACLE AMERICA, INC.,

Respondent.

**On Petition for a Writ of Certiorari to the
United States Court of Appeals
For the Federal Circuit**

**BRIEF OF *AMICUS CURIAE* RED HAT, INC. IN
SUPPORT OF PETITIONER**

ANDREW P. BRIDGES

Counsel of Record

FENWICK & WEST LLP

801 CALIFORNIA STREET

MOUNTAIN VIEW, CA 94041

(415) 875-2389

abridges@fenwick.com

DAVID L. HAYES

FENWICK & WEST LLP

555 CALIFORNIA STREET,

12th FLOOR

SAN FRANCISCO, CA 94104

(415) 875-2411

dhayes@fenwick.com

Counsel for Amicus Curiae Red Hat, Inc.

February 25, 2019

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	iv
INTEREST OF THE <i>AMICUS CURIAE</i>	1
STATEMENT.....	3
SUMMARY OF THE ARGUMENT	6
ARGUMENT	9
I. THE FEDERAL CIRCUIT’S DECISIONS UPSET SETTLED EXPECTATIONS ABOUT THE SCOPE OF COPYRIGHT AND ABOUT FREEDOM TO BUILD COMPATIBLE AND INTEROPERABLE SYSTEMS.	9
A. Compatibility and Interoperability Through Programming Interfaces Are Essential Features of Countless Digital Products and Services.	9
B. The Use of Others’ Interfaces, Including for Compatibility and Interoperability, Without the Need to Ask Permission and Secure Copyright Licenses, Is Ubiquitous and Essential.....	12

TABLE OF CONTENTS - Continued

	Page
C. A Legal Consensus That Interfaces Are Not Copyrightable, Resting Upon the Explicit Language of Section 102(b), Has Emerged Over Decades.	15
D. The Federal Circuit’s Disruption of the Legal Consensus Concerning the Free Use of Interfaces Causes Uncertainty Across Many Industries and Threatens Innovation.....	18
II. THE FEDERAL CIRCUIT’S DECISIONS MISAPPLIED THE COPYRIGHT ACT.....	19
A. Programming Interfaces Are “Necessary Incidents” for Other Software to Make Functional Use of the Programs’ Capabilities.....	19
B. The Federal Circuit’s Decisions Show That the Fair Use Doctrine Is Not an Acceptable Substitute for a Bright-Line Rule Against Copyright Control over Interfaces.	20

TABLE OF CONTENTS - Continued

	Page
C. The Federal Circuit's Decisions Improperly Allow Creators of Software Programs to Use Copyright to Control Unfairly All Uses of Products or Technology That Utilize Their Interfaces.	22
CONCLUSION.....	23

TABLE OF AUTHORITIES

CASES	PAGE
<i>Baker v. Selden</i> , 101 (11 Otto) U.S. 99 (1880)	<i>passim</i>
<i>Computer Assocs. Int’l, Inc. v. Altai, Inc.</i> , 982 F.2d 693 (2d Cir. 1992).....	15, 16
<i>Feist Publ’ns, Inc. v. Rural Tel. Serv.</i> Co., 499 U.S. 340 (1991)	16
<i>Lotus Dev. Corp. v. Borland Int’l, Inc.</i> , 49 F.3d 807 (1st Cir. 1995), <i>aff’d by</i> <i>an equally divided court</i> , 516 U.S. 233 (1996)	11, 16, 20
<i>Oracle Am., Inc. v. Google Inc.</i> , 750 F.3d 1339 (Fed. Cir. 2014)	18
<i>Whelan Assocs, Inc. v. Jaslow Dental</i> <i>Lab., Inc.</i> , 797 F.2d 1222 (3d Cir. 1986).....	15, 16
STATUTES	
Copyright Act	8, 19
17 U.S.C. § 102(b)	<i>passim</i>
OTHER AUTHORITIES	
U.S. Const. art. 1, § 8, cl. 8	2

INTEREST OF THE *AMICUS CURIAE*¹

Red Hat, Inc., with headquarters in Raleigh, North Carolina, is the world's leading provider of open source software and related services to enterprise customers. Using a community-powered approach to software development, Red Hat has a strong reputation for delivering reliable and high-performing cloud, Linux, middleware, storage, and virtualization technologies. Wall Street investment firms, hundreds of Fortune 500 companies, and the United States government use its products. Red Hat has more than 95 offices in over 35 countries.

Red Hat has a stake in the consistent and correct determination of the scope of copyright protection that applies to interfaces of computer programs, including the Java interfaces at stake in this case. It relies on the availability of open interfaces in developing new products, including products that are compatible with or interoperate with other computer products, platforms and services. Interoperability is the very foundation of the Internet, the Web, and of countless devices and services that depend upon them.

¹No person other than the *amicus curiae* and its counsel, including parties to this action and their counsel, authored this brief in whole or in part or contributed money that was intended to fund preparing or submitting this brief. Counsel of record for the parties received timely notification of the intent to file the brief and have consented to the filing of the brief.

Computer program code generally deserves copyright protection. Red Hat supports and relies on that protection in its own business. But it sees the Federal Circuit's decisions below as disturbing a well-established statutory principle and settled expectations upon which it and many other companies in information and communication technology and related service industries have built businesses for over two decades. In turn, the decisions create uncertainty for companies in countless other industries, like Red Hat's customers, that depend upon a vibrant environment for technological and business advancements. Contrary to the Constitutional purpose of copyright law, see U.S. Const. art. 1, § 8, cl. 8, the decisions of the Federal Circuit will chill the "Progress of Science and useful Arts."

Red Hat joins countless other companies that are watching this case in the hope that the Court will remedy the uncertainty and fear that the Federal Circuit has caused.

STATEMENT

Red Hat joins other companies that have grave concerns about the decisions of the Federal Circuit, which affect all types of software, devices, and services that require or rely upon interoperability. Red Hat is a leader in the development of Free and Open Source Software (“Open Source Software”). Open Source Software development today produces enterprise-grade software used by the world’s largest businesses. It is quickly becoming the preferred way to build new applications and create new functionality for existing applications.

The development of Open Source Software involves a wider pool of developers who are able to foster increased innovation. When compared to the employees of a single developer, this wider pool is likely to be more diverse and, with differing experiences, can lead to better software solutions. The involvement of a large base of developers also fosters greater efficiency, creating higher quality code faster and more effectively than a single company’s development team.

Open Source Software development has been one of the great technological success stories of recent decades. The Open Source Software collaborative development model is responsible for the creation of Linux (including Red Hat Enterprise Linux), which runs a large portion of the modern Internet. The popular WordPress content management system is developed under a similar collaborative open source

model. The rapid pace of innovation of the type described above is enabled by the ability to use interfaces that are unrestricted by copyright protection.

Open Source Software development depends upon both the power and the limits of copyright law. Copyright law gives Open Source Software developers the power to control certain downstream uses of their software code through use of various forms of Open Source Software licenses. But the law sets limits on that control that are vital to the freedom to interoperate. The Federal Circuit's distortion of both the power and limits of copyright, contrary to Section 102(b) of the Copyright Act and the broad understanding throughout industry, creates potential obstacles to further Open Source Software innovation.

If copyright were to subsist in software interfaces and the Open Source Software development communities were to have to adapt to the Federal Circuit's decisions with respect to the free availability and use of programming interfaces, they would face significant barriers in the creation and implementation of new modules to replace their current modules. This consequence could chill the innovation that is generated by Open Source Software community development and the progress of science and innovation that copyright law is supposed to promote.

As described above, computer programs achieve compatibility and interoperability with each other through a multitude of specifically defined interfaces.

The use of computer program interfaces for compatibility and interoperability purposes is both ubiquitous and essential to the operation of information and communication technologies and infrastructures. This fact has become even more so in today's ever more highly networked world. The freedom to utilize, implement, re-implement, and extend existing interfaces has been a key to competition and progress in the computer, information technology, communication technology, and networking fields.

Recognition that programming interfaces are uncopyrightable does not jeopardize copyright in software programs generally or deprive copyright owners of rewards for their creative authorship. Software developers create value, and differentiate themselves in the marketplace, through their implementation code. The more freely that programming interfaces are available, the more of a market may exist for particular implementations using the interfaces. This benefits both the developer of the original implementation and developers of other programs that may interact with the implementation, making all the interacting products more valuable. This is true whether the implementation is closed-source, with the developer getting royalties from those who adopt it, or is open source with the developer getting revenues from auxiliary services.

SUMMARY OF THE ARGUMENT

Over the last thirty years, despite some authority to the contrary, a strong judicial consensus emerged, resting upon the language of Section 102(b), that the statutory monopoly of copyright does not extend to programming interfaces. This consensus, which drew from cases going back to the Court's landmark copyright decision in *Baker v. Selden*, 101 (11 Otto) U.S. 99 (1880), unleashed a tidal wave of innovations in personal and mobile computing, cloud computing, e-commerce, and Internet services. Those innovations have, in turn, fueled a remarkable wave of business model innovations that have advanced science and medicine, stimulated the economy, and enhanced the well-being of individuals.

Both Federal Circuit decisions below call into question a bedrock legal assumption that has fostered tremendous innovation. The understanding that the copyright statutory monopoly does not extend to programming interfaces, and that others are free to use those interfaces, has caused innovation to flourish across the entire range of interoperable systems and networks. It has been fundamental to the Internet as we know it.

The Federal Circuit's decisions turn this commonly accepted understanding of the law on its head. They therefore threaten disruption across technology and communications industries including individual developers, small companies, and companies of substantial size such as Red Hat. The

decisions will cloud the rules for product development within the framework of copyright law and call into question approaches to software development in the marketplace today that countless developers created in reliance on long expectations that programming interfaces were freely available.

The Federal Circuit's first decision, by treating compatibility and interoperability as relevant only to fair use and not to copyrightability, would require a developer to perform a fair use analysis before developing virtually every compatible or interoperable product. This would be impractical, if not impossible. Fair use is a notoriously fact-specific doctrine, requiring case-by-case analysis, and it is no substitute for the bright-line rule that Section 102(b), 17 U.S.C. § 107(b), establishes. The Federal Circuit's second decision, overturning a well instructed jury's verdict on fair use, throws that doctrine into chaos. A developer must anticipate both how a rational jury would view very technical facts and how an appellate court, more removed from the facts, might lightly overturn a verdict.

The Federal Circuit's decisions are not just harmful but also profoundly incorrect. The first decision, upholding copyrightability of programming interfaces that facilitate communication between software systems and enable interoperability of many different technologies and services, ignored the careful contours of the statutory copyright monopoly that specifically exclude procedures, processes, systems, and methods of operation from copyright protection in Section 102(b) of the Copyright Act. The

second decision, overruling a jury that had found fair use after its careful examination of the facts of the case and correct application of the law, turned the entire framework of fair-use decision making on its head.

The Federal Circuit's treatment of fair use in its second decision exposes more clearly the error of its first decision. Freedom to build interoperable systems in this connected world must not turn on highly variable assessments of individual fact patterns by both juries and appellate courts that happen to disagree with the juries. The bright line of Section 102(b) both establishes an important boundary between copyright and patent law and provides reliable guidance to all.

Red Hat is concerned that the Federal Circuit's decisions will hand some copyright holders a patent-like veto power that the Copyright Act did not provide, and that Congress did not envision: the ability of a copyright holder to control the operations of *others'* products merely because they use its programming interface as a method for communicating or interoperating with the copyright holder's product. The Federal Circuit's dramatic shift in the scope of copyright protection for interfaces will be that technology will become less interoperable and therefore more fragmented, less standardized, and less useful, all to the detriment of technical progress and efficiency.

ARGUMENT

I. THE FEDERAL CIRCUIT'S DECISIONS UPSET SETTLED EXPECTATIONS ABOUT THE SCOPE OF COPYRIGHT AND ABOUT FREEDOM TO BUILD COMPATIBLE AND INTEROPERABLE SYSTEMS.

A. Compatibility and Interoperability Through Programming Interfaces Are Essential Features of Countless Digital Products and Services.

Computer programs are quintessential examples of utilitarian works. They are not pleasure reading, like novels. They often exist for the sole purpose of performing functions like carrying out financial transactions, monitoring home security and appliances, monitoring network traffic to detect and thwart threats or attacks, videoconferencing, coordinating family schedules, making travel reservations, and countless other useful tasks, and for powering hardware devices that persons use to perform those functions.

To perform their practical functions, computer programs must be compatible with, or interoperate with, other computer programs. Devices running software programs need to work with other devices running other software programs: for example, a modern smartphone must be able to help its owner

upload a photo to a social media site, send it to a friend by email to view on the friend's device, or send it to a printer. The number of possible combinations of individual devices and their software programs across the Internet is incalculably high. Convergence on common communication and interaction methods for devices and their software programs has powered the open Internet. Virtually all software and consumer product developers depend on interoperability, both furnishing and relying upon standard programming interfaces to facilitate synergy.

One computer program is “compatible” with a second computer program if the first program conforms to a set of commands, formats or rules utilized by the second program. A programming interface (also called, among other things, a “software interface,” or an “application programming interface” or “API”) is a defined symbolic mechanism by which a software program specifies its functions and by which other programs can “invoke,” or trigger, its operation.²

² Persons use a variety of terms in this context, and terms do not always have agreed meanings. Whatever the specific terminology, the point is that all programs and devices that communicate or interact with each other require tools to facilitate the actions, just as a toaster needs an interoperable plug and electrical outlet combination to allow it to connect to a standard power source (a point other *amici* have made). Convergence of both plugs and outlets on standard methods of connecting appliances to power benefits progress, competition, and consumer interests. Limits on technologies facilitating interoperability, if at all, are the province of *patent*, not *copyright*, law.

The landmark case of *Lotus Development. Corp. v. Borland International, Inc.*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided court*, 516 U.S. 233 (1996), involved two compatible computer programs. Lotus developed one of the first widely successful spreadsheet programs known as “Lotus 1-2-3.” Lotus 1-2-3 allowed users to store useful sequences of spreadsheet commands known as “macros” for accomplishing repeated tasks. Over time, a base of millions of users of Lotus 1-2-3 developed, many of whom had written macros that were critical to their business or personal use of Lotus 1-2-3.

The defendant Borland sought to develop a competing spreadsheet product that would have superior functionality. Borland believed, however, that it was unlikely to convince existing users to switch to its competing product unless Borland’s product was “compatible” with Lotus 1-2-3 in the sense that it would be able to read and execute existing macros with which persons were already familiar. In addition, because existing users had invested substantial time and effort in learning the Lotus commands, to compete, Borland’s program also needed to allow users to write new macros using familiar Lotus commands. These two compatibility features of Borland’s program—the capability to write macros using Lotus 1-2-3 commands and to read and execute existing Lotus macros—formed the basis of Lotus’s claim of copyright infringement against Borland.

In a related vein, a computer program must very frequently interoperate with another computer

program by exchanging information with the other program or by invoking one or more functions or operations available in the other program. For example, an application program such as a photo editor invokes functions, such as opening a file, storing a file, or printing a document, that the operating system can perform. The photo editing program must invoke those functions using precisely defined commands and syntax that the operating system interface requires, analogous to the Lotus 1-2-3 macros described above. More broadly, a photo editing program also needs to operate with printer software, display software, and email software; email software must operate with Internet network software and other email software or email services. These examples barely begin to describe the interoperation of countless devices and software programs of billions of persons across the Internet. Over the past 40 years the revolutions in personal and mobile computing, as well as the rise of the Internet, have greatly magnified the need for interoperability among different computer programs and devices running them.

B. The Use of Others' Interfaces, Including for Compatibility and Interoperability, Without the Need to Ask Permission and Secure Copyright Licenses, Is Ubiquitous and Essential.

The mechanism by which computer programs achieve compatibility and interoperability with each

other is through various specifically defined interfaces. The term “interface” broadly encompasses a wide range of technical mechanisms by which commands, data or other information are input into, or exchanged between, computer programs. For example, the commands of a program such as Adobe Photoshop are a type of interface, which the user uses to operate the program. Similarly, an “application programming interface” or API (such as the Java API at issue in this case) specifies the formats and rules (such as function names, parameters and inputs) by which one computer program (such as an application program running on a desktop computer) can trigger functions or operations in another computer program (such as a program running on a server in the cloud). A data file format is yet another type of interface that specifies the types and ordering of data that a computer program requires.

The use of computer program interfaces, including for compatibility and interoperability, without the need to ask permission and secure a copyright license, is both ubiquitous and essential to operation of information technologies and the communication infrastructure worldwide. Geometrically accelerating interconnectivity has led to the global interconnection of devices of virtually limitless types, such as home appliances, heart monitoring implants, embedded biochips in animals, security devices and clothing sensors. The leading technology industry analysis and consulting firm Gartner estimates there will be 14.2 billion Internet-

connected things this year. Free and open use of programming interfaces makes that possible.

The freedom to utilize, implement, re-implement, and extend existing interfaces has been the key to competition and progress in the computer, information technology, and networking fields since their beginning. It has enabled the development and advancement of the personal computer, operating systems, Open Source Software, programming languages, the Internet, and cloud computing. Compatible interfaces enable users to switch platforms (for example, from one operating system to another) or services (for example, from one cloud computing service provider to another) and avoid being locked in to their existing technology or service providers. Compatible interfaces also enable a service to be more widely available to users of different devices, such as smartphones or computers running different operating systems. Similarly, the ability to interoperate freely with other programs, devices, or services through their interfaces can enable the creation of innovative applications that expand the functions of social media, smartphones and tablets, medical devices, automobiles, kitchen appliances, home thermostats and security systems, payment cards, and countless other products across the digital landscape.

C. A Legal Consensus That Interfaces Are Not Copyrightable, Resting Upon the Explicit Language of Section 102(b), Has Emerged Over Decades.

Innovation occurs within frameworks of legal rules. Businesses make investments in, and calculate the risk of, innovation in reliance on those legal rules. Over the last 30 years, despite some authority to the contrary, a consensus understanding has developed in the computer and information technology industries, and in other industries that depend upon those technologies, that programming interfaces do not have copyright protection and therefore are free for anyone to use without needing to seek permission by negotiating for copyright licenses.

The foundation of that consensus is Section 102(b) of the Copyright Act, which states that “[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.” 17 U.S.C. § 102(b).

In the United States, judicial recognition of the non-copyrightability of interfaces began with the Second Circuit’s rejection in *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), of the Third Circuit’s decision in *Whelan Associates, Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986). Whelan had afforded a very broad

scope of copyright protection to essentially any aspect of a computer program other than its overall function defined at the highest level.

In an influential decision, the Second Circuit in *Computer Associates* concluded that the Court's decision in *Feist Publications, Inc. v. Rural Telephone Service. Co.*, 499 U.S. 340 (1991), undercut *Whelan*, and it ruled that under Section 102(b) copyright did not extend to program elements that are necessary for compatibility. 982 F.2d at 711. The First Circuit reached a similar outcome in *Lotus*, where the court ruled that the menu commands and command structure of Lotus 1-2-3, which Borland reproduced in its competitive spreadsheet program to achieve compatibility, constituted an uncopyrightable method of operation under Section 102(b). 49 F.3d at 815-18.

The judicial consensus that emerged from the *Computer Associates* and *Lotus* decisions derived and drew support from the Court's landmark copyright decision in *Baker*, which both the *Computer Associates* and *Lotus* decisions cited extensively. In *Baker*, Selden sought to protect a system of double entry bookkeeping by the copyright he held in a book describing the system. The book included certain forms or blanks, consisting of ruled lines and headings, illustrating the system and showing how it was to be used and carried out in practice. Selden asserted that, because the ruled lines and headings were part of the book, copyright protected them and no one could copy or use similar ruled lines and headings without violating his copyright. 101 U.S. at 100- 01.

The Court rejected Selden's argument, ruling that the copyright on a work "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, so as to prevent an engineer from using them whenever occasion requires." *Id.* at 103. An exclusive right to a method of operation or system could be secured, if at all, only through a patent. *Id.* at 102-03. The Court therefore concluded that where "the art" (the system or method of operation) taught by a copyrighted work "cannot be used without employing the methods and diagrams used to illustrate" the art in the work, "or such as are similar to them, such methods and diagrams are to be considered as *necessary incidents to the art, and given therewith to the public . . . for the purpose of practical application.*" *Id.* at 103. (emphasis added). In other words, the system or method of operation described or embodied in a copyrighted work is not copyrightable, and the "necessary incidents" required to practice the system or method of operation are also not copyrightable (they are "given . . . to the public").

Under *Baker*, it does not matter to copyrightability whether the "necessary incidents" could be written in another way or the original author exercised choice in creating them. As "necessary incidents" to a system or method of operation, they are not copyrightable. The Federal Circuit failed to take account of this fundamental rule of *Baker* when it ruled that "copyrightability is focused on the choices available to the [author] at the time the computer

program was created.” *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1370 (Fed. Cir. 2014).

D. The Federal Circuit’s Disruption of the Legal Consensus Concerning the Free Use of Interfaces Causes Uncertainty Across Many Industries and Threatens Innovation.

Based on the preceding legal authorities and others, all industries that rely upon digital technologies have come to a decades-long understanding and expectation that programming interfaces are available for everyone to use in creating new products and services. Countless companies, in reliance on that understanding, have unleashed a tidal wave of innovation in personal and mobile computing, cloud computing, e-commerce, finance, health care, security, and other Internet- and device-enabled fields. The Federal Circuit’s decisions turn this widespread understanding of the law on its head. If the Court allows the decisions to stand, they will provoke uncertainty among developers, small companies, and large companies alike. Red Hat is no exception, which motivates it to submit this brief.

The Federal Circuit’s rulings have inflicted uncertainty on the development of software; on the design of connected devices, features, and services; and on programs and devices that interact with other programs. Anxious eyes from all corners of the globe await the outcome of this case. If the Court does not grant review and correct the Federal Circuit’s failure

to respect Section 102(b) and that court's departure from settled expectations, the resulting uncertainty may impede an unknowable number of innovative projects. The uncertainty could also provide fertile ground for predatory litigation, forcing innovators to circle their wagons to defend their *existing* products (which have relied upon decades of legal consensus) instead of forging new technology and commerce paths. It may lead to inefficient and inferior "silo" development or impose transaction costs for product interoperability. That could impede the vast benefits of technological and economic synergy that broad and free interoperability has provided up to now. The deleterious effect of the Federal Circuit's departure from settled law on innovation, competition, and everyday life in the connected world is foreseeable and deserves the Court's attention.

II. THE FEDERAL CIRCUIT'S DECISIONS MISAPPLIED THE COPYRIGHT ACT.

A. Programming Interfaces Are "Necessary Incidents" for Other Software to Make Functional Use of the Programs' Capabilities.

As functional works, virtually all computer programs embody a system or method of operation in the form of functional capabilities of the programs, and the programs' interfaces are among the "necessary incidents" required to make use of such

functional capabilities. *Cf. Baker*, 101 U.S. at 103. For that reason, the interfaces are methods of operation, embodied in symbolic language, that Section 102(b) places outside the scope of copyright protection. For example, in *Lotus*, the First Circuit cited and analogized to *Baker* to support its conclusion that the Lotus menu command hierarchy and macro language (its interface for human interaction and programmed operation) were uncopyrightable: “Lotus wrote its menu command hierarchy so that people could learn it and use it. Accordingly, it falls squarely within the prohibition on copyright protection established in *Baker v. Selden* and codified by Congress in § 102(b).” 49 F.3d at 817.

B. The Federal Circuit’s Decisions Show That the Fair Use Doctrine Is Not an Acceptable Substitute for a Bright-Line Rule Against Copyright Control over Interfaces.

By treating compatibility and interoperability as relevant only to fair use, and not to copyrightability, the Federal Circuit’s first decision would require a developer to perform a fair-use analysis before developing a compatible or interoperable product. This is unworkable. Fair use is a notoriously fact-specific doctrine, requiring case-by-case analysis. The litigation risk of an error in a technology developer’s fair-use analysis is great. That risk alone, when it involves major investments in fundamental industry practices, may operate as a

persistent brake upon innovation and investments. Instead, clear application of the well-recognized statutory limitation in Section 102(b) is essential. Otherwise, the uncertainty caused by the decisions below will impede innovation in technology areas requiring reproduction or other uses of interfaces for compatibility or interoperability.

The Federal Circuit's disturbing second decision shows precisely why its first decision was wrong. The second decision, overturning a well-instructed jury's verdict, disregards jurors' conclusions after they deeply considered the evidence and applied the law. For the Federal Circuit first to urge deference to a jury's assessment of the fact-specific question of fair use, and then to ride roughshod over the jury's assessment, makes any prediction of fair-use outcomes impossible. Technology developers cannot innovate in an environment where standards and outcomes are uncertain and where they must—because of the decisions in this case—survive an unpredictable, high-stakes litigation gauntlet to secure a stable position for their innovations in the global marketplace and economy.

C. The Federal Circuit's Decisions Improperly Allow Creators of Software Programs to Use Copyright to Control Unfairly All Uses of Products or Technology That Utilize Their Interfaces.

The fair use doctrine is not a sufficient bulwark against threats against future innovators who seek to create compatible or interoperable programs using existing interfaces. It is too indeterminate, and its outcomes are too unpredictable, as the Federal Circuit's erroneous overturning of the jury's verdict shows. The correct bulwark lies in the bright-line rule that the Copyright Act provides: Section 102(b) precludes that type and degree of control over methods of operation, which include programming interfaces. Copyright protection is not equivalent to patent protection. Nor should it be, with the much longer duration of copyright and the lack of rigorous standards or meaningful examination in the process of copyright registration.

The result of the Federal Circuit's unwarranted expansion of copyright law, contrary to Section 102(b) of the Copyright Act, together with its failure to respect a proper jury verdict on fair use, will be technology and communications infrastructures, systems, and services that are less interoperable, and therefore more fragmented, less standardized, and less useful, all to the detriment of "Progress of Science and useful Arts."

There is much more to say about the errors of the Federal Circuit, but that can await briefing on the merits. For the present, however, Red Hat emphasizes the urgency and importance of a resolution of the issues. Red Hat therefore urges the Court to grant a writ of certiorari on both questions that the petition presents.

CONCLUSION

For decades the legal rules establishing the non-copyrightability of programming interfaces have been a foundation of innovation and competition in this country and throughout the world. The erroneous decisions below have thrown industry expectations and the legal rules into disarray. The copyright issues that the petition raises deserve immediate attention, and Red Hat urges the Court to grant Google's petition on both questions that it presents.

Respectfully submitted,

ANDREW P. BRIDGES
Counsel of Record
 FENWICK & WEST LLP
 801 CALIFORNIA STREET
 MOUNTAIN VIEW, CA 94041
 (415) 875-2389
 abridges@fenwick.com

DAVID L. HAYES
 FENWICK & WEST LLP
 555 CALIFORNIA STREET
 12th FLOOR
 SAN FRANCISCO, CA 94104
 (415) 875-2411
 dhayes@fenwick.com

Counsel for Amicus Curiae Red Hat, Inc.

February 2019