No. 14-410

IN THE

Supreme Court of the United States

GOOGLE, INC.,

Petitioner,

v.

ORACLE AMERICA, INC.,

Respondent.

ON PETITION FOR WRIT OF CERTIORARI
TO THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

BRIEF OF *AMICUS CURIAE* PUBLIC KNOWLEDGE IN SUPPORT OF THE PETITION

CHARLES DUAN
Counsel of Record
SHERWIN SIY
PUBLIC KNOWLEDGE
1818 N Street NW, Suite 410
Washington, DC 20036
(202) 861-0020
cduan@publicknowledge.org

Counsel for amicus curiae

TABLE OF CONTENTS

TABLE OF AUTHORITIES

CASES

CONSTITUTIONAL PROVISIONS

STATUTES

OTHER SOURCES

INTEREST OF AMICUS CURIAE

Public Knowledge is a non-profit organization that is dedicated to preserving the openness of the Internet and the public's access to knowledge, promoting creativity through balanced intellectual property rights, and upholding and protecting the rights of consumers to use innovative technology lawfully. As part of this mission, Public Knowledge advocates on behalf of the public interest for a balanced copyright system, particularly with respect to new and emerging technologies.[1]

Public Knowledge has previously served as *amicus* in key copyright cases. *E.g.*, *Kirtsaeng v. John Wiley & Sons, Inc.*,, 133 S. Ct. 1351 (2013); *Golan v. Holder*, 132 S. Ct. 873 (2012); *Moseley v. V Secret Catalogue, Inc.*, 537 U.S. 418 (2003); *Eldred v. Ashcroft*, 537 U.S. 186 (2003).

[1]Per Supreme Court Rule 37(6), no counsel for a party authored this brief in whole or in part, and no counsel or party made a monetary contribution intended to fund the preparation or submission of the brief. No person or entity, other than *amicus*, its members, or its counsel, made a monetary contribution to the preparation or submission of this brief. Per Rule 37(2)(a), counsel of record for all parties received notice of intent to file this brief at least 10 days prior to the due date, and both parties granted consent to filing. Documents indicating such are being filed herewith.

SUMMARY OF ARGUMENT

Say that Delphi Corporation manufactures screws. It hits upon a new design for a screw socket—the interface between screw and screwdriver—that is more efficient than the prevailing Phillips and flathead insertions. Capitalizing on this novel idea, Delphi manufactures a line of screws using this socket, which it calls Sumatra.

The Sumatra socket is wildly popular. New lines of screwdrivers are made for the Sumatra socket. Engineering textbooks praise the Sumatra design. Woodworkers teach their sons and daughters to use it. And competing screw manufacturer Zillion decides to make its own screws compatible with the Sumatra socket. The screws otherwise differ, but use the Sumatra socket so that woodworkers need not purchase new tools.

Only then does Delphi declare the Sumatra socket a sculptural work, suing Zillion for copyright infringement.

Software programs are today's screws and screwdrivers of computer engineering. Just as screws and screwdrivers require a common socket in order to work, software requires a common language—an application programming interface. And just as Delphi may not leverage copyright to block competition in screw socket shape, Oracle in the present case may not use copyright to monopolize the Java system interface.

Copyright is granted to promote the public interest in generating new creative works, and as such balances between securing incentives for authors and ensuring an open space of ideas upon which future creators may build. As part of that balance, this Court and others have repeatedly held, since the venerable *Baker v. Selden*, that

no copyright may inhere in functional elements of a work. Such elements, being incidents of practical knowledge and the useful arts, cannot be restrained under a copyright system intended for aesthetic expression.

The Federal Circuit ignored this essential mandate, by finding copyrightable an element of software that is quintessentially a method of operation. Left uncorrected, that decision threatens to undermine the fundamental balance of copyright law.

But it also threatens to undermine the decades of progress in Internet and computer technology that have come about only because the openness of technology interfaces enabled enormous competitive growth. To sanction copyright, not on the workings of a computer program, but *on the ways of using that program*, would be to sanction monopolization of those fundamental elements that enabled such competitive growth.

This case thus presents a question of national importance on an issue of federal policy. This Court should grant certiorari to correct a decision that would otherwise stray far from longstanding precedent. This Court should grant certiorari to ensure that the progress of science and the useful arts is not slowed.

And this Court should grant certiorari to protect the constitutional balance of copyright law, now upset by a decision too zealous in expanding private rights at the expense of the public good. For what hangs in that balance is not merely a small corner of computer science. What hangs in the balance are those basic ideas that, as this Court once declared, must remain "free to all men and reserved exclusively to none."

ARGUMENT

It is textbook law that functional elements of a work cannot be copyrighted. 17 U.S.C. § 102(b) (2013) (copyright protection does not extend "to any idea, procedure, process, system, method of operation, concept, principle or discovery"); *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992) ("To the extent that a work is functional or factual, it may be copied . . . ."); 1 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 2.18(d)(2) (2014) [hereinafter Nimmer]. Yet the Court of Appeals for the Federal Circuit extracted the one element of a computer program that is quintessentially functional—the application programming interface—and declared it protected by copyright.

Left unchecked, the Federal Circuit's decision sets a precedent that threatens not only to topple the careful balance of copyright law mandated by the Constitution and calibrated by this Court and Congress, but also to severely hamper the progress of innovation for decades to come. The Court should grant certiorari to correct this grave error of tremendous national importance.

I. This Case Presents a Question of Importance to the Future of Technology

Prior to reviewing the substantial legal errors of the decision below, this Court must appreciate the magnitude of the problem created by those errors, to understand the substantiality of the federal question for which this Court should grant certiorari. Thus, this section proceeds in two parts: first, to explain the nature of Oracle's copyright claim; and second, to review how that claim will severely harm the progress of technology.

A. Oracle Essentially Claims Copyright in a Way of Communicating with a Computer

At the heart of the case is the concept of the "application programming interface," or API. Oracle charges Google with copying the API of the Java system, and the Federal Circuit's analysis focused on two components of that API, the "declaring code" and its structure, sequence and organization. *See Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1347, 1356 (Fed. Cir. 2014). Thus, the precise meaning of that term is essential to this case.

Oracle calls the term a "verbal chameleon," Opening Brief and Addendum of Plaintiff-Appellant at 9, *Oracle* (Feb. 11, 2013) (No. 13-1021) [hereinafter Oracle Brief], but in fact it has a uniform, understandable definition.[2] **An application programming interface is a set of commands and rules for communicating with a computer program.** Technical dictionaries define "API" as an "interface through which one program can communicate with another." Editors of the Am. Heritage Dicts.,

[2]It is in fact only Oracle's mistaken usage of the term "API" that creates any confusion. Oracle uses "Java API" to refer to an entire bundle of computer programs, not just declaring code. *See id.* This does not conform with the general understanding of the term "API," but Oracle repeatedly insists on conflating its own mistaken definition with the correct one. *See* Response and Reply Brief of Plaintiff-Appellant at 5, *Oracle* (July 3, 2013) (No. 13-1021). Oracle does in fact admit the correct definition, Oracle Brief, *supra*, at 9 (API can describe a "communication protocol to pass information between programs"), and then invents the term "declaring code" to mean "API," *see id.* at 10 (declaring code is "code that the programmer declares in order to invoke the prewritten program").

For purposes of clarity, this brief uses the term "Java API" to refer only to the declaring code and its organizational structure; the term "Java system" will refer to the entire bundle of computer programs.

Dictionary of Computer and Internet Words 11 (2001); *accord A Dictionary of Computing* 19 (Oxford Univ. Press 6th ed. 2008) (defining API as "a set of functions and procedures [that] enables a program to gain access to facilities within an application"). The D.C. Circuit has defined the term as "routines or protocols that perform certain widely-used functions" that are "expos[ed]—i.e., ma[de] available to software developers." *United States v. Microsoft Corp.*, 253 F.3d 34, 53 (D.C. Cir. 2001).

This definition should not sound foreign, because the concept of an "interface" is well known from many fields. A "user interface" includes the windows, icons, and other graphical elements by which people communicate with computers. *See, e.g., Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1438 (9th Cir. 1994) (describing the Apple Macintosh user interface as "a user-friendly way for ordinary mortals to communicate with the Apple computer"); Andries van Dam, *Post-WIMP User Interfaces*, Comm. ACM, Feb. 1997, at 63, *available at* http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.6390.[3] Outside the computer context, "interface" can mean "communication or interaction," or "a thing or circumstance that enables separate and sometimes incompatible elements to coordinate effectively." *Random House Unabridged Dictionary* 993 (2d ed. 1987).[4]

---

[3] WIMP stands for "Windows, Icons, Menus, and mouse Pointer."

[4] *See also Rios v. Colon*, 819 F.2d 319, 328 (1st Cir. 1987) (describing government official as "a kind of cultural interface" because duties included acting as "a public liaison between the government and the artistic community"); *Trans-Lux Corp. v. United States*, 696 F.2d 963, 964 (Fed. Cir. 1982) (calling a device designed to enable communication between a Telex network and a user terminal an "interface between" the two).

The Java API comprises lines of "declarations." As one example, the Java system can compute the larger of two numbers. This capability may be used through the following API command (explained further in Figure 1):

public static int *max*(**int** *x*, **int** *y*)

Oracle, 750 F.3d at 1349. The declaration—which exemplifies what Oracle is claiming copyright in—serves as a template for the programmer to fill in. To use the command, a programmer would follow the template to write:

max(3, 4)

and the Java system would return as output the larger value, namely 4. *See id.* at 1350.

	[D]	[C]	[A]	[B]
DECLARATION:	**public static int**		*max* (**int** *x*, **int** *y*)	
EXAMPLE USE:			*max* (	3, 4)

[A] *This is the name of command.*

[B] *These are the template for providing inputs to the command. Here, the command expects two inputs of integers.*

[C] *This prefix indicates that the command output will be an integer.*

[D] *These two prefix words indicate that the API command can be called by anyone in any context.*

Figure 1: Example Java API command and usage

Two main conclusions about application programming interfaces deserve mention.

First, as stated initially, *an API is simply the commands and rules for communicating with a computer program*, as precise and exacting as those rules may be. A programmer uses the Java API to operate the Java system, much as a remote control car driver uses knobs and buttons to operate the car.

Second, *an API is a generalized concept, not a particular writing or piece of code*. It is much like a language or protocol. It may be embodied in a work *using* it or a dictionary *enumerating* its vocabulary. *See* James Gosling et al., *The Java Application Programming Interface* xviii (1996). But standing alone, it is merely abstract knowledge, enabling two parties to understand and perform the wishes of each other.

B. To Extend Copyright to Basic Computer Communication Methods Would Rapidly Decelerate Innovation

Although the application programming interface is a relatively straightforward concept, it is of critical importance to modern innovation. APIs underlie practically every significant advance in computing technology, and permitting copyright to extend over those APIs would severely hinder further advances in that technology.

The history of computer technology is marked by incremental progress, with scores of innovations emanating out of individual, well-designed platforms—each presenting an interface to downstream innovators. Microsoft Windows famously maintained a consistent API

for many years, allowing for the creation of numerous Windows software programs. *See* Ian Murdock, *On the Importance of Backward Compatibility*, Ian Murdock's Weblog (Jan. 14, 2007), http://ianmurdock.com/platforms/on-the-importance-of-backward-compatibility/.

Furthermore, the success of the Internet has been credited to its "use of a common protocol," a single language with which all contemporary computers can communicate. Paul E. Ceruzzi, *A History of Modern Computing* 295–96 (2d ed. 2003). Every web page owes its existence to the HyperText Transport Protocol by which computers obtain web pages. *See* T. Berners-Lee et al., *Hypertext Transfer Protocol—HTTP/1.0* (1996), *available at* http://tools.ietf.org/html/rfc1945. Each of these interfaces—Windows, the Internet, HTTP—served as a springboard for enormous further advancement of technology.

If copyright circumscribed the ability of others to use and emulate these interfaces, it would have balkanized software and strangled the Internet.[5] The Federal Circuit dismissed any problems with copyright on APIs because others "could have chosen different ways to express and implement the functionality." *Oracle*, 750 F.3d at 1368. But following this mandate for operating sys-

[5]Oracle makes a halfhearted attempt to distinguish its case from a general copyright grant on computer interfaces. *See* Response and Reply Brief of Plaintiff-Appellant at 5, *Oracle* (July 3, 2013) (No. 13-1021). But its attempts to appeal to the ambiguity of the term "API"—an ambiguity that, as noted at note 2 *supra* p. 5, was manufactured by Oracle itself—are unavailing. Oracle offers no reason why its arguments as to the copyrightability of "declaring code" (its term for an API) are at all distinguishable from any other interface, computer or otherwise.

tems would have led to a proliferation of incompatible operating system interfaces, creating a headache for software developers—one that even Oracle criticized.[6] *Cf.* Neil Harper, *Server-Side GPS and Assisted GPS in Java* 69 (2010) (explaining how a lack of an available standard for a particular GPS location function led to "a proliferation of proprietary interfaces"). And following that mandate would have turned the Internet from a universal information resource into a Tower of Babel, every website speaking a different API language.

This is problematic in its own right, but particularly problematic in view of the purpose of copyright to promote the progress of science and the useful arts. *See* U.S. Const. art. 1, § 8, cl. 8. This Court has repeatedly held that "copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work." *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349–50 (1991) (citing *Harper & Row Publishers, Inc., v. Nation Enters.*, 471 U.S. 539, 556–57 (1985)).

Sanctioning copyright in the Java API contravenes this principle: it would allow Oracle to unilaterally block further public use and innovation. Copyright is not "a game of chess in which the public can be checkmated" in this way. *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 679 (1st Cir. 1967).

---

[6]Oracle noted that, prior to Java, "computer programmers had to pick one platform when writing new programs," and touted that Java "allows programmers to write programs that run on different types of computer hardware." Oracle Brief, *supra*, at 8 (alterations omitted).

II. The Decision Below Misapprehends the Role of Functionality in Copyright

Besides being bad policy, the Federal Circuit's decision is erroneous on the law. As the following discussion demonstrates, the Federal Circuit has decided an important federal question of copyrightability of functional elements, in a way that conflicts with the relevant decisions of this Court. Certiorari is warranted to correct that error.

A. Oracle's Copyright Claim Is Entirely Foreclosed by *Baker v. Selden*

Blank forms are generally not copyrightable because they are functional. *See Baker v. Selden*, 101 U.S. 99, 101–02 (1880). Application programming interfaces are similarly not copyrightable because they are merely the result of taking a blank form and saying "apply it with a computer." *Cf. Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2358 (2014).[7]

[7]Although it may appear that *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) (per curiam), considered and rejected this argument, that case actually contemplated something quite different. *Lotus* considered whether menu items in a spreadsheet application were methods of operation and thus not copyrightable. *See id.* at 813. Borland argued that the case was controlled by *Baker* because the spreadsheet functions of the application were analogous to the blank forms of *Baker*. *See Lotus*, 49 F.3d at 814. Unsurprisingly, the court dismissed this argument, since the menu items, and not the spreadsheet functions, were at issue. *See id.* The First Circuit never considered whether the menu items themselves were on par with the blank forms of *Baker*.

The elements of blank forms are identical to the elements of an API. Using the identifiers of Figure 1 on page 7, the Java API can be understood as a collection of declarations of commands, each declaration comprising [A] an identifier for the command, [B] a template of inputs to be provided with the command, and [C] a specification of outputs or results caused by performing the command. And, indeed, a blank form is composed of [A] a name identifying the form and its purpose, [B] a series of blank spaces forming a template for form inputs, and [C] an intended purpose or result to be achieved when the form is submitted.

This identity of content leads to identity of function. A blank form enables structured communication between a person filling in the form and a person reading the filled form. For example, Selden's condensed ledger enabled a bookkeeper to efficiently communicate financial records to an auditor or other reviewer. *See* Charles Selden, *Selden's Condensed Ledger and Condensed Memorandum Book* 8 (1861) [hereinafter *Selden's Condensed Ledger*], *available at* http://lcweb2.loc.gov/service/rbc/rbc0001/2011/2011gen155867/2011gen155867.pdf.[8]

The blank form thus serves as an interface between the writer and reader. An API is an interface as well; the only difference is that the reader is a computer rather than a person. *Cf. Ross, Brovins & Oehmke, PC v. Lexis/Nexis Group*, 463 F.3d 478, 485 (6th Cir. 2006) ("automation" of filling in forms "not sufficiently original to

[8]This book contains no page numbers, so the numbers identified in the citations refer to the page numbers of the PDF document cited in the URL.

warrant copyright protection").

Oracle repeatedly points to the "intricate web of connections" of the Java API, in an effort to suggest that its structure, sequence and organization of the API is copyrightable. Oracle Brief, *supra*, at 26. But so too can uncopyrightable blank forms constitute an intricate web of connections. Selden's book included 19 forms and 24 pages of demonstrative explanation designed "to compress almost innumerable accounts under a few specific, intelligible heads." *Selden's Condensed Ledger*, *supra*, at 8. For either blank forms or APIs, intricacy does not confer copyrightability.

Given that an API is factually on par with a blank form, it is unsurprising that the reasoning of *Baker* directly applies to the copyrightability of APIs. *Baker* held that blank ledger forms, including the "ruled lines and headings," could not properly be the subject of copyright. 101 U.S. at 101–02. The Court said that copyright cannot cover "systems" or an "art"; the Java API is certainly a system, one that teaches the "art" of using the Java system. *See id.* at 102; *Oracle*, 750 F.3d at 1351 (describing the API as an "overall system of organized names" (quoting *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 999 (N.D. Cal. 2012))).

The Java API is on all fours with the blank forms of *Baker*, both factually and legally. Since copying of the blank forms in *Baker* was permissible, copying of the Java API is too. *Cf. Feist*, 499 U.S. at 361 ("Not all copying, however, is copyright infringement.") Accordingly, the Federal Circuit's decision directly contradicts *Baker*.

B. Whether an Element Is Functional Turns on the Nature of the Element's Use, Not the Existence of *Ex Ante* Alternatives

The Federal Circuit found the Java API copyrightable because "alternate expressions are available" for the API command names, so their choices for command names and inputs were creative and thus fully protected from copying.[9] But numerous cases have held elements of a work unprotectable by copyright, even though the authors made choices among alternatives in constructing the work.

Baker certainly supports the proposition that functionality is not obviated by the existence of *ex ante* alternatives. The Court held Selden's blank forms uncopyrightable, even though Selden himself recognized that one could alternatively reorganize the forms to "place

[9] *Oracle*, 750 F.3d at 1360 (quoting *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 840 (Fed. Cir. 1992)); *see id.* at 1367 ("[A]n original work—even one that serves a function—is entitled to copyright protection as long as the author had multiple ways to express the underlying idea."); *id.* at 1363 ("Because Oracle exercised creativity in the selection and arrangement of the method declarations when it created the API packages and wrote the relevant declaring code, they contain protectable expression that is entitled to copyright protection.").

Among other reasons, the Federal Circuit appeared to arrive at this conclusion from a logical misreading of *Feist*. In that case, this Court held that a minimal degree of creativity was *necessary* for copyright to inhere in a work. *See* 499 U.S. at 362. But the Federal Circuit took *Feist* to mean that a minimal degree of creativity is *sufficient* for copyright. *See Oracle*, 750 F.3d at 1354, 1362; *see also id.* at 1365 (distinguishing *Lotus Development Corp. v. Borland International, Inc.* on the grounds that the Java API is "creative and original").

the names of Accounts to the extreme left or right—or other than central—of the columns." *Selden's Condensed Ledger*, *supra*, at 35.

Cases in diverse fields have followed suit. Even with facts involving subjects as far apart as games,[10] basketball rules,[11] and wavy bicycle racks,[12] courts have uniformly found the subject matter not copyrightable and thus not infringed by copying, even though in each case the author of the work made creative choices among alternatives (terminology for games, names of basketball fouls, and alternate shapes of waves, respectively).

Most relevant to the present case, courts have held that names of commands used to control a computer program are not properly the subject of copyright, despite the fact that certainly "alternate expressions are available." *Lotus Development Corp. v. Borland International, Inc.* specifically held that the selection of names of menu items in the Lotus 1-2-3 computer program "is not copyrightable because it is part of Lotus 1-2-3's 'method

[10] *See, e.g., Anti-Monopoly, Inc. v. Gen. Mills Fun Group*, 611 F.2d 296, 300 n.1 (9th Cir. 1979) ("[B]usiness ideas, such as a game concept, cannot be copyrighted.") (citing *Baker*); *Affiliated Hosp. Prods., Inc. v. Merdel Game Mfg. Co.*, 513 F.2d 1183, 1188 (2d Cir. 1975) ("The rules of the game are perforce in the public domain as well as the game itself."); 1 Nimmer, *supra*, § 2.18[H][3] ("[N]o copyright may be obtained in the system or manner of playing a game . . . .").

[11] *See Official Rules of the National Basketball Association* R. 12 (2013–2014), *available at* http://mediacentral.nba.com/media/mediacentral/Official-NBA-Rule-Book.pdf (naming the various types of fouls); *Hoopla Sports & Entm't v. Nike, Inc.*, 947 F. Supp. 347, 354 (N.D. Ill. 1996) (citing 1 Nimmer, *supra*, § 2.18[H][3]).

[12] *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1146 (2d Cir. 1987) (holding that the wave shape's "aesthetic design elements are significantly influenced by functional considerations").

of operation.'" 49 F.3d 807, 816 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996) (per curiam). Similarly, *Computer Associates International v. Altai, Inc.* held that nearly all of the "parameter lists and macros" of a computer program were unprotected as being "either in the public domain or dictated by the functional demands of the program." 982 F.2d 693, 714 (2d Cir. 1992). Importantly, the macros and parameter lists, as defined in that case, correspond almost precisely to API commands and inputs claimed to be copyrightable here.[13]

Even though, in all these cases, there were aspects of the overall *work* that may have been original, the asserted *element* of each work was not copyrightable. "The mere fact that a work is copyrighted does not mean that every element of the work may be protected." *Feist*, 499 U.S. at 348.

Specifically, all of the above elements deemed unprotectable were functional, and of note, many of those unprotectable elements (game play, basketball rules, computer menu items) are in essence mechanisms of communication (with other game players, between referees and basketballers, between user and spreadsheet application). These examples are not far afield from the Java API, a functional mechanism of communication with the Java system.

Thus, the Federal Circuit greatly erred in relying heavily on the existence of *ex ante* alternatives to justify

[13] "A macro is a single instruction that initiates a sequence of operations or module interactions within the program," and a parameter list is "the information sent to and received from a subroutine." *Id.* at 697–98.

a finding of copyrightability. The decision misstates well-established law, and this Court should grant certiorari to prevent further propagation of that error.

III. The Federal Circuit Mischaracterizes the Interface of Copyright and Patent

In discussing "Google's policy-based arguments" on the respective domains of patent and copyright, the Federal Circuit misunderstands arguments that *interfaces* are more properly protected by patent law than copyright law. *See Oracle*, 750 F.3d at 1379–80. Google, several *amici* below, and the district court merely proffered the unremarkable argument that functional elements should be excluded from copyright by § 102(b) and the idea/expression dichotomy. *See, e.g., Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 984 (N.D. Cal. 2012) (questioning whether copyright grants "an exclusive right to a functional system"). But the Federal Circuit mistook them to mean that software may only be patentable or copyrightable, but not both. *See Oracle*, 750 F.3d at 1380 (considering arguments as to whether "software is or should be entitled to protection only under patent law—not copyright law," or "the exact opposite"). The Federal Circuit further assumed that criticisms of software patents equate to suggestions to expand copyrightable subject matter to cover interfaces. *See id.*

These propositions are flawed. First, the Federal Circuit neglects that there is matter outside the realm of *both* copyright and patent; the court apparently supposed that every element of a software program must fit into one or the other. Second, the Federal Circuit fails to differentiate the discrete elements of a given software product that

may be copyrightable and those that may be patentable, instead lumping those elements together into a single entity. Third, the Federal Circuit conflates programming interfaces with computer programs generally.

A. Abstract Ideas Are Neither Copyrightable Nor Patentable

Certainly a computer program may contain both copyrightable and patentable elements, and that software can be copyrighted. But it does not follow that every element of that computer program is copyrightable. *See, e.g.*, *Lotus*, 49 F.3d at 813–14; *Altai*, 982 F.2d at 703–06; *cf. Mazer v. Stein*, 347 U.S. 201, 217 (1954).

Indeed, it is well known that there is matter outside the scope of *both* copyright and patent. Laws of nature, natural phenomena, and abstract ideas may not be patented. *See, e.g., Alice*, 134 S. Ct. at 2354 (listing judicial exceptions to 35 U.S.C. § 101); *Funk Bros. Seed Co. v. Kalo Inoculant Co.*, 333 U.S. 127, 130 (1948) (deeming such elements unpatentable because they must remain "free to all men and reserved exclusively to none"). Nor may they be copyrighted: abstract ideas are definitionally excluded by the idea-expression dichotomy, and natural laws and phenomena lack the required creativity or a human author. *See, e.g.*, *Mazer*, 347 U.S. at 217; *Harper & Row Publishers, Inc., v. Nation Enters.*, 471 U.S. 539, 547 (1985); *Christianson v. W. Publ'g Co.*, 149 F.2d 202, 203–04 (9th Cir. 1945); *Ansehl v. Puritan Pharm. Co.*, 61 F.2d 131, 137–38 (8th Cir. 1932); *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930); *Feist*, 499 U.S. at 358–60; *Burrow-Giles Lithographic Co. v. Sarony*, 111 U.S. 53, 58 (1884) (finding photographs copyrightable

"so far as they are representatives of original intellectual conceptions of the author"); *Uranita Found. v. Maaherra*, 114 F.3d 955, 958 (9th Cir. 1997); *see also Bilski v. Kappos*, 130 S. Ct. 3218, 657 (2010) (Breyer, J., concurring) (questioning patentability of business methods); Nimmer, *supra*, § 2.18[H][3][c], n.70 (noting that business methods are not copyrightable).

The fact that there are clear examples of matter neither copyrightable nor patentable should demonstrate that a reduction in the scope of one need not lead to an expansion in the scope of another. If patent protections on software are reduced, as some commenters wish, that does not automatically raise copyright protection in tandem. Patent and copyright are not a seesaw; each is a separate doctrine independent of the other.

B. Potentially Patentable Processes are Not Themselves Copyrightable Expression

Arguments as to whether or not a particular product should be copyrightable or patentable are asking the wrong question; different aspects of any given product may receive different types of protection. For example, the lamps at issue in *Mazer* contained creative expression protected by copyright. 347 U.S. at 214. Lamps, as a general matter, are useful articles that, if novel and nonobvious, may be protected by patent. *See* 35 U.S.C. §§ 101–103 (2013). Yet the copyrightable elements of those lamps are not the same as the patentable elements.

Kieselstein-Cord v. Accessories by Pearl, Inc., 632 F.2d 989 (2d Cir. 1980), is illustrative. Where a designer

belt buckle contains both a functional and a creative element, courts analyzing copyrightability distinguish between those elements that serve the utilitarian function (holding the belt) and those that express the creative aspect. *Id.* at 991–93; 17 U.S.C. § 101. The fact that the same physical structures within the object simultaneously achieve both purposes does not prevent that severability in the legal analysis.

The same is true of any given artistic work. The ideas contained within (e.g., a pair of star-crossed lovers) are severable from their expression (e.g., within the specific expression of *West Side Story*). *See Reed-Union Corp. v. Turtle Wax, Inc.*, 77 F.3d 909 (7th Cir. 1996). Patentability of those ideas is an inquiry wholly separate from copyrightability of that expression.

The literary works that fall into the subcategory of computer programs are no different. Computer programs can embody utilitarian processes, abstract ideas, and creative expression simultaneously. *See, e.g., Altai*, 982 F.2d at 704. But the aesthetic elements of a computer program may not receive utility patents, and the utilitarian processes and ideas cannot be copyrighted.

The confusion of the part for the whole feeds into the policy argument raised by the Federal Circuit via citation to two news articles. *See Oracle*, 750 F.3d at 1380; *Stalking Trolls*, Economist, Mar. 8, 2014, *available at* http://www.economist.com/news/technology-quarterly/21598321-intellectual-property-after-being-blamed-stymying-innovation-america-vague; Timothy B. Lee, *Will the Supreme Court Save Us from Software Patents?*, Wash. Post: Switch Blog (Feb. 26, 2014), http://www.washingtonpost.com/blogs/the-switch/

wp/2014/02/26/will-the-supreme-court-save-us-from-software-patents/.

Both of these pieces were written prior this Court's decision in *Alice*, and they are written in view of that pending controversy. Accordingly, both articles consider patentability of software *as a whole*, and recommend copyright on software *as a whole*. They do not at all contemplate whether certain elements of software, such as abstract ideas or APIs, merit copyright protection standing alone. They do not contemplate those questions because they were not pertinent at the time, and because it is blackletter law that those elements are not copyrightable. The Federal Circuit's citation of those articles to support its view of copyrightability of APIs misreads not only the articles but also the law.

C. Arguments Regarding the Utility of Interfaces Are Not Addressed to Software as a Whole

Furthermore, the Federal Circuit confuses APIs with computer programs generally. For example, the decision below characterized Google's policy arguments as "premised on the belief that copyright is not the correct legal ground upon which to protect intellectual property rights to *software programs*." *Oracle*, 750 F.3d at 1379 (emphasis added). But to support that characterization, the court quotes Google's brief, that under Ninth Circuit precedent, "developers could no longer hope to protect *software interfaces* by copyright." *Id.* at 1380 (alterations omitted, emphasis added).[14]

[14]The Federal Circuit also mischaracterized the district court's opinion, *compare id. with Oracle Am., Inc.*, 872 F. Supp. 2d at 984

This failure to differentiate API from software is a critical error on the part of the Federal Circuit. As explained above, the level of expressiveness contained within an API is minimal at best—*because an API is not software, but rather a concept embodied within software*. *See* Section I.A *supra* p. 5. (The Federal Circuit's error, again, is likely attributable to Oracle's mistaken explication of the term "API." *See* note 2 *supra* p. 5.)

In fact, among the many articles the Federal Circuit cites, only one addresses software generally, as opposed to programming interfaces specifically: an article by Professor Pamela Samuelson written thirty years ago and cited by neither Google nor any *amici*. *See Oracle*, 750 F.3d at 67; Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 Duke L.J. 663. That article, evidently cited by Oracle in an attempt to marginalize Samuelson's later scholarship, *see* Response and Reply Brief of Plaintiff-Appellant at 22, *Oracle* (July 3, 2013) (No. 13-1021), acknowledged shortcomings in the copyright case law of 1984 regarding software copyrightability, *see* Samuelson, *supra*, at 754–56. This Court need not be reminded of the progress made in copyright and patent law in the intervening three decades.[15] Furthermore, the article did not advocate for software's classification as solely patentable material at all, but rather suggested a theoretical *sui generis* software protection.

(assessing whether "exclusive right to a *functional system, process, or method of operation* [] belongs in the realm of patents, not copyrights"), and several *amici*.

[15] *See, e.g.*, *Alice*, 134 S. Ct. at 2358; *Bilski*, 130 S. Ct. at 3231; *Feist*, 499 U.S. at 364; *Harper & Row*, 471 U.S. at 569; *Altai*, 982 F.2d at 706; *Lotus*, 49 F.3d at 814.

See id. at 762–69. As such, its relevance to the current case, or current policy discussions, is truly academic.

By mischaracterizing the functionality arguments against APIs as an attack on the copyrightability of software generally, the Federal Circuit converts a substantive argument about a narrow class of computer code into a strawman of a policy argument. Such fallacious reasoning should carry no weight before this Court.

* * *

No doubt there is a certain amount of sympathy to be had for Oracle, who repeatedly remarked, and inveigled the Federal Circuit to remark, on the amount of effort the creators of Java put into a complex software system.[16] But while it "may seem unfair that much of the fruit of the compiler's labor be used by others without compensation," that is not "some unforeseen byproduct of a statutory scheme" but rather "the essence of copyright."[17] Copyright demands a balance between private reward and public advancement. The present case threatens to tip that balance too far backward, and the Court, holding the fulcrum of that balance, must recenter it.

[16] *See, e.g.*, Oracle Brief, *supra*, at 8 ("It took Sun's most senior, experienced and talented developers years to write some of [the Java code]." (internal quotations omitted)); *Oracle*, 750 F.3d at 30 n.6 ("Sun/Oracle developers had to wrestle with what functions to include in [a Java] package . . . .").

[17] *Feist*, 499 U.S. at 349 (quoting *Harper & Row*, 471 U.S. at 589 (Brennan, J., dissenting)).

CONCLUSION

For the foregoing reasons, the petition for a writ of certiorari should be granted.

Respectfully submitted,

CHARLES DUAN
Counsel of Record
SHERWIN SIY
PUBLIC KNOWLEDGE
1818 N Street NW, Suite 410
Washington, DC 20036
(202) 861-0020
cduan@publicknowledge.org

Counsel for amicus curiae