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 PROFESSORS PETER S. MENELL, DAVID NIMMER, AND SHYAMKRISHNA BALGANESH AS *AMICI CURIAE* IN SUPPORT OF PETITIONER

PETER S. MENELL
KORET PROFESSOR OF LAW
Counsel of Record
UNIVERSITY OF CALIFORNIA,
AT BERKELEY SCHOOL OF LAW
225 Bancroft Way
Berkeley, CA 94720-7200
(510) 642-5489
pmenell@law.berkeley.edu

January 10, 2020

COCKLE LEGAL BRIEFS (800) 225-6964
WWW.COCKLELEGALBRIEFS.COM

TABLE OF CONTENTS

Page

INTEREST OF *AMICI* 1

SUMMARY OF ARGUMENT 2

ARGUMENT 4

I. THE JURISPRUDENTIAL AND LEGISLATIVE FRAMEWORK FOR INTERPRETING INTELLECTUAL PROPERTY PROTECTION FOR FUNCTIONAL FEATURES OF COMPUTER SOFTWARE 5

A. *Baker v. Selden* (1879) 8

B. The Modern Statutory Framework 10

II. APPLICATION PROGRAM INTERFACES (APIs), JAVA APIs, AND GOOGLE'S IMPLEMENTATION OF A PARTIALLY INTEROPERABLE MOBILE PLATFORM..... 14

A. Development of the Java Programming Environment 15

B. Google's Android Implementation 18

III. THE FEDERAL CIRCUIT'S DECISIONS CONFLICT WITH THIS COURT'S SEMINAL RULING IN *BAKER V. SELDEN*, MISINTERPRET CONGRESS'S CODIFICATION OF INTELLECTUAL PROPERTY LAW'S FUNDAMENTAL CHANNELING PRINCIPLE AND COPYRIGHT'S LIMITING DOCTRINES, AND UPEND WELL-REASONED AND LONG-STANDING JUDICIAL DECISIONS THAT HAVE SUPPORTED INNOVATION AND COMPETITION IN THE COMPUTER INDUSTRY 22

TABLE OF CONTENTS—Continued

Page

A. The Federal Circuit's Decisions Conflict with *Baker v. Selden* 23

B. The Federal Circuit's Decisions Misconstrue the Copyright Act 24

C. The Federal Circuit's Decisions Revive and Exacerbate Circuit Splits on Copyrightability, Merger, and Fair Use..... 27

D. Reversing the Federal Circuit's *Oracle v. Google* Decisions Maintains the Coherence of the Intellectual Property System and Will Restore Peace and Clarity to the Computer Software Industry 31

IV. GOOGLE INDEPENDENTLY IMPLEMENTED THE JAVA API FUNCTIONAL SPECIFICATIONS AND THEREFORE DID NOT INFRINGE ORACLE'S COPYRIGHTS 33

V. THE FAIR USE ISSUE IS MOOT..... 36

CONCLUSION..... 36

TABLE OF AUTHORITIES

Page

CASES

Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240 (3d Cir. 1983) 24, 26, 29

Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006 (N.D. Cal. 1992), *aff'd in part, rev'd in part*, 35 F.3d 1435 (9th Cir. 1994) 29

Baker v. Selden, 101 U.S. 99 (1879) ..... *passim*

Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1989) 31

Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) 20, 29

Dastar Corp. v. Twentieth Century Fox Film Corp., 539 U.S. 23 (2003) 31

Eng'g Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335 (5th Cir. 1994) 29

Gates Rubber v. Bando Chem. Indus., Ltd., 9 F.3d 823 (10th Cir. 1993) 29

Lexmark Int'l, Inc. v. Static Control Components, Inc., 387 F.3d 522 (6th Cir. 2004) 29, 30

Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996) 29

Mazer v. Stein, 347 U.S. 201 (1954) 25

MiTek Holdings, Inc. v. ARCE Engineering Co., 89 F.3d 1548 (11th Cir. 1996) 29

Mitel, Inc. v. Iqtel, Inc., 124 F.3d 1366 (10th Cir. 1997) 29

TABLE OF AUTHORITIES—Continued

Page

Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930) 29

Oracle Am., Inc. v. Google Inc., 750 F.3d 1339 (Fed. Cir. 2014) .....*passim*

Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) 19, 23, 32, 36

Plains Cotton Coop. Assoc. v. Goodpasture Comput. Serv., Inc., 807 F.2d 1256 (5th Cir. 1987).....29

Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).....20, 29, 34

Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (1936) 20

Sony Comput. Entertainment v. Connectix, 203 F.3d 596 (9th Cir. 2000).....20, 29

TrafFix Devices, Inc. v. Marketing Displays, Inc., 532 U.S. 23 (2001) 31

Vault Corp. v. Quaid Software Ltd., 847 F.2d 255 (5th Cir. 1988).....24

Wal-Mart Stores, Inc. v. Samara Brothers, Inc., 529 U.S. 205 (2000) 31

Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986) 27, 28, 29

CONSTITUTIONAL PROVISION

U.S. Const., Art. I, § 8, cl. 8 31

TABLE OF AUTHORITIES—Continued

Page

STATUTES

17 U.S.C. § 101 .....11, 12

17 U.S.C. § 102(a)(1) 11

17 U.S.C. § 102(b).....*passim*

17 U.S.C. § 117 12

35 U.S.C. § 100(b).....9

1902 Design Patent Act, Act of May 9, 1902, ch. 783, § 4929, 32 Stat. 193 (1902).....26

Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3007, 3028 (1980) 12

Act of Dec. 31, 1974, Pub. L. No. 93-573, § 201, 88 Stat. 1873 (1974) 11, 12

Copyright Act of 1976, Pub. L. No. 94-553, 90 Stat. 2541 ..... 11, 24, 25, 26

RULE

Sup. Ct. R. 37.6 1

OTHER AUTHORITIES

Android GPS [Google Product Strategy]: Key strategic decisions around Open Source (July 26, 2005) 19

Application programming interface, WIKIPEDIA, https://en.wikipedia.org/wiki/Application_programming_interface 19

TABLE OF AUTHORITIES—Continued

Page

BAKER'S REGISTER OF RECEIPTS AND DISBURSEMENTS WITH BALANCE SHEETS AND REPORTS FOR COUNTY AUDITORS AND TREASURERS 8

Brian Profitt, The Impact of Oracle's Defense of API Copyrights, ITWORLD (Aug. 23, 2011), http://www.itworld.com/article/2738675/mobile/the-impact-of-oracle-s-defense-of-api-copyrights.html 30

CHARLES SELDEN, SELDEN'S CONDENSED LEDGER, OR BOOK-KEEPING SIMPLIFIED (1859) 8

David Bank, The Java Saga, WIRED (Dec. 1, 1995) 16

Declaration, WIKIPEDIA https://en.wikipedia.org/wiki/Declaration_(computer_programming) 35

Donald S. Chisum, Rochelle Cooper Dreyfuss, Paul Goldstein, Robert A. Gorman, Dennis S. Karjala, Edmund W. Kitch, Peter S. Menell, Leo J. Raskind, Jerome H. Reichman & Pamela Samuelson, LaST Frontier Conference on Copyright Protection of Computer Software, 30 JURIMETRICS J. 15 (1989) 28

H.R. REP. NO. 82-1923 (1952) 9

H.R. REP. NO. 94-1476 (1976) 11, 12, 24

Jason J. Du Mont & Mark D. Janis, The Origins of American Design Patent Protection, 88 INDIANA L.J. 837 (2013) 25

TABLE OF AUTHORITIES—Continued

Page

Michael Hussey, Copyright Captures APIs: A New Caution for Developers, TECHCRUNCH (Nov. 3, 2015), https://techcrunch.com/2015/11/03/copyright-captures-apis-a-new-caution-for-developers/ 30

NAT'L COMM'N ON NEW TECH. USES OF COPYRIGHTED WORKS, FINAL REPORT 1 (1979) 12

NIMMER ON COPYRIGHT

§ 13.03[F] 28

§ 2.01 6

§ 2A.05[A][2][a] 9

§ 2A.06[A][1] 11, 12

§ 2A.07[B][2] 10

§ 2A.07[D][4][b] 25

§ 8.08[A][3] 24

§ 9.10[A][1] 6

P. Anthony Sammi, Christopher A. Lisy & Andrew Gish, Good Clean Fun: Using Clean Room Procedures in Intellectual Property Litigation, 25 INTELL. PROP. & TECH. L.J. 3 (2013) 20

Peter S. Menell & Suzanne Scotchmer, Economic Models of Innovation: Stand-Alone and Cumulative Creativity, in BEN DEPOORTER & PETER S. MENELL (EDS.), RESEARCH HANDBOOK ON THE ECONOMICS OF INTELLECTUAL PROPERTY LAW, VOLUME 1: THEORY 119 (2019) 6

TABLE OF AUTHORITIES—Continued

Page

Peter S. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 STAN. L. REV. 1045 (1989) 28

Peter S. Menell, An Epitaph for Traditional Copyright Protection of Network Features of Computer Software, 43 ANTITRUST BULL. 651 (1998).....30

Peter S. Menell, API Copyrightability Bleak House: Unraveling and Repairing the *Oracle v. Google* Jurisdictional Mess, 31 BERKELEY TECH. L.J. 1515 (2016).....22

Peter S. Menell, Economic Analysis of Network Effects and Intellectual Property, 34 BERKELEY TECH. L.J. 218 (2019).....13

PETER S. MENELL, MARK A. LEMLEY, & ROBERT P. MERGES, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGY AGE: 2019, Vol. I.....5

Peter S. Menell, Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software, 31 HARV. J.L. & TECH. 303 (2018).....*passim*

Ruggles Design Bill, S. 269, 26th Cong. § 1 (1841)..... 26

S. REP. NO. 94-973 (1975).....12

Thomas B. Hudson, A Brief History of the Development of Design Patent Protection in the United States, 30 J. PAT. & TRADEMARK OFF. SOC'Y 380 (1948) 26

U.S. Patent No. 271,363 (1883).....6

INTEREST OF *AMICI*[1]

The authors of this brief are professors of law who study and teach intellectual property law. Their interest in filing this brief is to promote faithful interpretation of U.S. copyright law.

Peter S. Menell is the Koret Professor of Law and co-founder and director of the Berkeley Center for Law & Technology (BCLT) at the University of California at Berkeley. He holds a law degree and a doctorate degree in economics. Professor Menell has authored or co-authored more than 100 articles and authored, co-authored, or edited 15 books, including leading casebooks, intellectual property treatises (including sections of NIMMER ON COPYRIGHT), and research handbooks.

David Nimmer is Of Counsel at Irell & Manella, LLP and Professor from Practice at the UCLA School of Law, where he regularly teaches copyright law and related subjects. Since 1985, he has authored NIMMER ON COPYRIGHT, maintaining up-to-date the treatise originally published in 1963 by his late father, Melville B. Nimmer.

---

[1] Pursuant to Sup. Ct. R. 37.6, *amici* represent that 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 this brief. No person other than *amici* made a monetary contribution to its preparation or submission. Petitioner and Respondent have consented in writing to the filing of this brief and were given 10 days notice of *amici*'s intent to file.

Shyamkrishna Balganesh is Professor of Law and Co-Director of the Center for Technology, Innovation & Competition (CTIC) at the University of Pennsylvania Law School. His research and teaching focus on the interplay of intellectual property law, innovation policy, and the common law. He co-authored the portions of NIMMER ON COPYRIGHT relating to *Baker v. Selden*, 101 U.S. 99 (1879).

In Spring 2018, the HARVARD JOURNAL OF LAW & TECHNOLOGY published a Special Issue on copyright protection for computer software focusing on *Oracle v. Google*. The issue is framed by Professor Menell's monograph-length lead article *Rise of the API Copyright Dead?*, which explores the rich history, technology, and legal issues surrounding this case. The Special Issue includes commentaries prepared by counsel from both sides of the litigation as well as leading academics. Professor Nimmer's treatise, NIMMER ON COPYRIGHT, has been cited extensively in the decisions below as well as throughout copyright jurisprudence.

SUMMARY OF ARGUMENT

The Federal Circuit's decisions in *Oracle v. Google* conflict with this Court's seminal decision in *Baker v. Selden*, 101 U.S. 99 (1879), misinterpret Congress's codification of this Court's fundamental channeling principle and related limiting doctrines, and upend nearly three decades of sound, well-settled, and critically important decisions of multiple regional circuits on the scope of copyright protection for computer

software. Based on the fundamental channeling principle enunciated in *Baker v. Selden*, as reflected in § 102(b) of the Copyright Act, the *functional requirements* of APIs for computer systems and devices, like the internal workings of other machines, are outside of the scope of copyright protection even as non-merged aspects of the implementing code for APIs are protectable. Google *independently implemented* the functional specifications of the 37 APIs at issue and hence did not infringe Oracle's copyrights.

By way of brief illustration, copyright protects artistic and literary works, such as a creative metal sculpture or haiku. Nonetheless, the proprietors of those works cannot complain when third parties replicate elements of that expression that are essential to the operation of a particular machine. For instance, a car manufacturer could secure the ignition switch for its automobiles via a metal key with an original cut pattern on the blade. Although that pattern might be protected as a modern sculpture, the car manufacturer could not use copyright law to prevent others from utilizing the same expression for the purpose of starting the car. The same consideration applies to a car manufacturer that secures a digital ignition switch via entry of a haiku. Copyright law does not bar third parties from utilizing the necessary expression of that otherwise protectable literary work for the purpose of starting the car. The computer program implementing that digital key may be protected by copyright law, but the law places no bar on copying the essential functional elements needed to operate the ignition switch—the

haiku text and any other indispensable functional features of the computer program.

As *Baker v. Selden* recognized, copyright law's limiting doctrines implement a constitutional and statutory balance intended to promote progress by channeling functional features exclusively to the utility patent regime. Although copyright can protect separable expressive features, such as surface ornamentation of an ignition key or non-merged implementing code of a digital ignition key, it does not bar the use of *functional specifications*—the essential technological elements. Only utility patent law can protect those features.

ARGUMENT

As background for addressing copyright protection for the works at issue in this case, Section I reviews the jurisprudential and legislative framework governing functional features of works of authorship. Section II then explains the pertinent technical background relating to the Java application program interfaces (APIs) and Google's implementation of a subset of the APIs. With this background in place, Section III exposes the Federal Circuit's critical errors and explains that *Baker v. Selden* and the Copyright Act enable software companies to develop interoperable and partially interoperable software programs so long as they *independently implement* prior developers' *functional specifications*. It also discusses how the

Federal Circuit's approach upends well-reasoned and long-standing judicial decisions that have supported innovation and competition in the computer industry. Applying the proper interpretation of copyright protection for APIs to this case, Section IV confirms that Google's independent implementation of functional specifications of the 37 APIs at issue did not infringe Oracle's copyrights. As a result, there is no need for the Court to reach the fair use question.

I. THE JURISPRUDENTIAL AND LEGISLATIVE FRAMEWORK FOR INTERPRETING INTELLECTUAL PROPERTY PROTECTION FOR FUNCTIONAL FEATURES OF COMPUTER SOFTWARE

The intellectual property system has long addressed the interplay of the different modes of protection by establishing a clear hierarchical structure to ensure fidelity to Congress's goals in promoting both technological and creative progress. The Patent Act requires the inventor or discoverer of a process, machine, manufacture, or composition of matter to meet stringent threshold requirements of novelty, non-obviousness, and disclosure in order to obtain 20 years of protection. *See* PETER S. MENELL, MARK A. LEMLEY, & ROBERT P. MERGES, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGY AGE: 2019, Vol. I, at 16-26, 36-37, 156-68. In this way, patent protection aims to promote pioneering and cumulative innovation by providing a relatively strong form of protection for a relatively short

duration. *See* Peter S. Menell & Suzanne Scotchmer, Economic Models of Innovation: Stand-Alone and Cumulative Creativity, in BEN DEPOORTER & PETER S. MENELL (EDS.), RESEARCH HANDBOOK ON THE ECONOMICS OF INTELLECTUAL PROPERTY LAW, VOLUME 1: THEORY 119, 120-50 (2019). The Copyright Act, by contrast, merely requires that authors surmount a low originality threshold to obtain relatively long-term protection—now life of the author plus 70 years. *See* NIMMER ON COPYRIGHT §§ 2.01, 9.10[A][1] (2019). But because copyright protection is limited to the author's original expression and does not extend to any ideas, procedures, processes, methods of operation, concepts principles or discoveries, it has far less impact on competition.

It would be nonsensical for copyright protection to encompass functional features of works of authorship, such as the gears and levers of a cash register (a sculptural work) or the functional specifications for a digital cash register (computer software). Cash registers are machines that fall squarely within the province of utility patent protection and have long been protected accordingly. *See*, *e.g.*, U.S. Patent No. 271,363 (1883) (Cash Register and Indicator). If copyright protection extended to functional specifications of machines or processes, a sculptor could protect the precise configuration of gears and levers for a mechanical cash register or a computer programmer could protect the functioning of a particular digital cash register (perhaps a spreadsheet) merely by meeting copyright's low originality threshold, thereby

circumventing the patent system's high protection thresholds and relatively short duration.

This Court's seminal decision in *Baker v. Selden*, later codified in the text and reflected in the legislative history of the Copyright Act, ensures that such illogical circumvention cannot occur. The channeling principle is clear: where functionality and expression inextricably coincide, copyright cannot subsist. Such elements can be protectable only if they satisfy patent law's higher thresholds, and that protection lasts only 20 years.

This principle does not deny the creativity of functional sculpture and computer system design features. Those elements may be highly creative. But when the function and expression merge—even in a very specific configuration—copyright protection must give way to ensure the coherence of the intellectual property system. The idea-expression dichotomy, merger doctrine, and other limiting doctrines (such as scenes a faire) implement a constitutional and statutory balance intended to promote progress by channeling functional features exclusively to the utility patent regime. Although copyright can protect separable expressive features, such as surface ornamentation of a physical cash register or non-merged implementing code of a digital cash register, it cannot extend to *functional specifications*—the essential technological elements. Only utility patent law can protect those features.

A. *Baker v. Selden* (1879)

This Court enunciated the structural foundation for the U.S. intellectual property system in *Baker v. Selden*, 101 U.S. 99 (1879). Accountant Charles Selden devised a condensed ledger bookkeeping system for government accounting. *See* CHARLES SELDEN, SELDEN'S CONDENSED LEDGER, OR BOOK-KEEPING SIMPLIFIED (1859). His new system consolidated the broad range of county transactions into a single ledger. The preface to Selden's book proclaimed that this new system would "greatly simplify the accounts of extensive establishments doing credit business" and handle "an almost infinite variety of transactions," qualifying it "to be classed among the greatest benefactions of the age." *Id*. It noted that "[i]n addition to the copyrights of this little book, [Selden] has applied for a patent right to cover the forms of the publication, and prevent their indiscriminate use by the public." *Id*.

In 1867, another accountant released BAKER'S REGISTER OF RECEIPTS AND DISBURSEMENTS WITH BALANCE SHEETS AND REPORTS FOR COUNTY AUDITORS AND TREASURERS, offering a similar accounting system with some advantages that made it easier to use. By 1871, Baker's system was in wide use while Selden's languished.

Selden's widow sued Baker for copyright infringement. On appeal, this Court recognized that Selden sought to monopolize use of the accounting system or method explained in the book through copyright law, thereby gaining an exclusive right in the use of similar

ruled lines and headings. *See Baker v. Selden*, 101 U.S. at 101. While acknowledging that copyright law protected an author's expression in conveying information on the subject of bookkeeping, the Court ruled that copyright protection could not extend to the "art"[2] or methods thus described:

> To give to the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters-patent, not of copyright. The claim to an invention or discovery of an art or manufacture must be subjected to the examination of the Patent Office before an exclusive right therein can be obtained; and it can only be secured by a patent from the government.

Id. at 102; *see also* NIMMER ON COPYRIGHT § 2A.05[A][2][a]. Justice Bradley illustrated the proposition by noting that although a physician could gain a copyright in a book about his medical discoveries and treatments, protection could not extend to the new art, manufacture, or composition of matter described in the book absent a utility patent. *Id*. at 103-04. Likewise,

> 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

---

[2] In modern parlance, "art" refers to process. *See* 35 U.S.C. § 100(b); H.R. REP. NO. 82-1923, at 6 (1952).

> employs to explain them, so as to prevent an engineer from using them whenever occasion requires. The very object of publishing a book on science or the useful arts is to communicate to the world the useful knowledge which it contains. But this object would be frustrated if the knowledge could not be used without incurring the guilt of piracy of the book. And where the art it teaches cannot be used without employing the methods and diagrams used to illustrate the book, 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; not given for the purpose of publication in other works explanatory of the art, but for the purpose of practical application.

Id. at 103-04. *Baker v. Selden* thereby established the fundamental principle for channeling protection among the intellectual property regimes. *See* NIMMER ON COPYRIGHT § 2A.07[B][2].

B. The Modern Statutory Framework

By the mid-1970s, the question of whether copyright protection extends to computer software emerged as Congress was putting the finishing touches on the overhaul of the 1909 Copyright Act. Rather than further delay completion of a legislative process that had been gestating nearly two decades, Congress established the National Commission on New Technological Uses of Copyrighted Works ("CONTU") to study the implications of the new technologies and recommend

revisions to federal intellectual property law. Act of Dec. 31, 1974, Pub. L. No. 93-573, § 201, 88 Stat. 1873 (1974).

As a stopgap, Congress included computer software within the scope of "literary works," one of the classes of "works of authorship" covered by the Copyright Act of 1976. *See* Copyright Act of 1976, Pub. L. No. 94-553, 90 Stat. 2541; 17 U.S.C. § 102(a)(1). The House Report explains that

> [t]he term 'literary works' does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. It also includes *computer data bases*, and *computer programs* to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.

H.R. REP. NO. 94-1476 at 53-54 (1976) (emphasis added). Other provisions of the 1976 Act endorsed *Baker v. Selden* and codified traditional exclusions for ideas and functional features. *See* 17 U.S.C. § 102(b) ("In 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."); *id*. at § 101 (definition of "pictorial, graphic, and sculptural works" excludes "mechanical or utilitarian aspects"); *see also* NIMMER ON COPYRIGHT

§ 2A.06[A][1]. The legislative history further explains that

> *Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer*, rather than merely to the 'writing' expressing his ideas. 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*.

H.R. REP. NO. 94-1476 at 57 (emphasis added); S. REP. NO. 94-973, at 54 (1975) (same).

After conducting extensive hearings and receiving expert reports, a majority of CONTU concluded that the intellectual work embodied in computer software should be protected under copyright law, but subject to the fundamental principle that copyright cannot protect "any idea, procedure, process, system, method of operation, concept, principle, or discovery" and the Supreme Court's foundational decision on the idea-expression dichotomy in *Baker v. Selden*. *See* NAT'L COMM'N ON NEW TECH. USES OF COPYRIGHTED WORKS, FINAL REPORT 1 (1979) (referencing 17 U.S.C. § 102(b)) ("CONTU REPORT"). CONTU recommended two changes to the 1976 Act, which Congress duly implemented. Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3007, 3028 (1980) (codified at 17 U.S.C. §§ 101, 117 (2012)). Most

importantly, the CONTU REPORT confirmed that while "one is always free to make a machine perform any conceivable process (in the absence of a patent), . . . one is not free to take another's program," subject to copyright's limiting doctrines. *See* CONTU REPORT at 20 (footnote omitted). It further explained that:

> The 'idea-expression identity' exception provides that copyrighted language may be copied without infringing when there is but a limited number of ways to express a given idea. This rule is the logical extension of the fundamental principle that copyright cannot protect ideas. In the computer context this means that when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement.

Id. (footnote omitted).

Thus, while recognizing important limitations on copyright protection for computer software, including the § 102(b) limitations, Congress intended that software programmers would garner protection for their program design and coding choices to the extent that the expression was separable from the underlying ideas and functionality. That is, the *functional specifications* required for a computer to operate in a particular way would remain outside the scope of copyright protection while the programmer's implementation of such specifications would be eligible for copyright

protection. The general programming ideas, particular requirements to achieve any function, and unoriginal programming choices remain free for others to use while the creative effort in particularized implementing code would gain protection. In this way, copyright would stand in the way of piracy and slavish copying without interfering with competition in machine functionality.

II. APPLICATION PROGRAM INTERFACES (APIs), JAVA APIs, AND GOOGLE'S IMPLEMENTATION OF A PARTIALLY INTEROPERABLE MOBILE PLATFORM[3]

An API is a software interface or communication protocol between different parts of a computer program or system intended to simplify implementation and maintenance of the software, control access to the software or hardware, and/or facilitate development of interoperable applications. *See* Application programming interface, WIKIPEDIA; Peter S. Menell, Economic Analysis of Network Effects and Intellectual Property, 34 BERKELEY TECH. L.J. 218, 224-30 (2019) ("Network Effects and IP"). An API can provide a feature or software library for a software system. APIs can thus function as high-level programming short-cuts that allow programmers to more easily create applications that

[3] This Section draws from Peter S. Menell, Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software, 31 HARV. J.L. & TECH. 303, 347-75 (2018) ("Rise of the API Copyright Dead").

will interoperate with a system. Rather than programming a feature from scratch, a programmer can access a pre-programmed feature by invoking the pertinent API function calls. Thus, programmers can more easily develop applications to run on a system by learning the API function calls. *See id*. at 224 ("The technical standards governing access to platforms, commonly referred to as application program interfaces (APIs) in the software industry, play a critical role in consumer and programmer adoption decisions, market entry, and competition.").

API packages function as the gears and levers of a virtual machine. Declarations or function calls serve as keys to unlock computing methods and operations.

A. Development of the Java Programming Environment

In 1990, a small programming team at Sun Microsystems set out to develop a new general programming language. The project evolved into an effort to create a programmable device for television set-top boxes. When Sun failed to interest consumer electronics or cable companies, Sun co-founder Bill Joy realized that the effort could be re-purposed to program webpages, the tantalizing new computing environment gaining salience in Silicon Valley.

The team soon produced "Java," a simple, lean, platform-independent, real-time, embeddable, multitasking programming language for web functionality. It used a similar syntax to the popular C programming

language, but was more compact, efficient, and secure. It enabled programmers to write "Java applets" (small application programs) that could run on Apple, Windows, or UNIX machines without customization. It accomplished this versatility through the use of the Java Virtual Machine ("JVM"), an intermediate layer of software that checks the code and insulates end users' computers from crashes and errors. Java enabled real-time interactivity, multimedia, and animation, which greatly enhanced the dynamism of webpages.

Following the "'profitless' approach to building market share" that Netscape had employed in giving away its Navigator browser, Sun made the Java programming language freely available. *See* David Bank, The Java Saga, WIRED (Dec. 1, 1995) (quoting Bill Joy, "There was a point at which I said, 'Just screw it, let's give it away.' Let's create a franchise."). As part of its effort to establish Java as the standard programming language for the Internet and critical to its efforts to prevent Microsoft from undermining Java's "Write Once, Run Anywhere" interoperability, Sun successfully pursued an open development path. *See id*.

Sun rolled out the first stable Java Development Kit in early 1996 and continued to expand features over the following year. The Java language comprises words, symbols, and pre-written programs to carry out various commands, such as printing something on the screen or performing a basic mathematical calculation. Sun organized sets of pre-written programs (methods, which are grouped in classes) into API packages (or class libraries). Each API package reflects a set of

declarations or functional specifications needed to invoke the methods; it functions as a mini-machine for performing particular pre-programmed functions. It is executed through detailed implementing code. Although Java programmers can write new code (methods) from scratch, the pre-written methods within the Java API packages provide convenient, efficient, reliable, standardized building blocks, thereby saving programmers tremendous time and effort.

In 1998, Sun released the Java 2 Standard Edition ("SE") platform. It contained eight API packages, three of which—java.lang, java.io, and java.util—were necessary to use the Java programming language. Sun gradually expanded the number of API packages, classes, and methods. Sun also established the Java Community Process to enable users to participate in the development of standard technical specifications for Java technology. By December 2006, the Java SE 6 platform contained over 100 APIs. Although Sun made the Java programming language freely available, it licensed the APIs under the General Public License ("GPL") that required users to make available any software incorporating the licensed code on a "share and share alike" basis. *See* Network Effects and IP at 227-28, 260-63. Sun also required licensees to ensure interoperability with the entire Java platform. *See* Rise of the API Copyright Dead at 355.

B. Google's Android Implementation

As smartphones emerged in the early 2000s, Google came to see that an open source platform for mobile communications was critical to enabling mobile devices to provide full web browsing capability. Building an open mobile communications platform, however, posed substantial challenges. A new operating system would need to be optimized for the small chips on which handsets were based. The devices would have to work in real-time. The platform had to be compact and optimized to the particular functionalities consumers would demand. In addition, the licensing model had to balance openness with downstream competition and innovation. Google did not believe that the GPL would provide sufficient flexibility for the range of players needed to establish a robust new mobile platform. In particular, Google worried that the viral share and share alike provision would discourage telecommunications companies and handset manufacturers (original equipment manufacturers ("OEMs")) from making investments in innovative features. A more permissive licensing model, in which downstream suppliers could build proprietary extensions on top of the base platform, would better promote robust competition and innovation. *See* Network Effects and IP at 263-64.

Google's Android team also believed that they would need to create an application programming environment that was familiar and easy to use. Java, with its wide adoption and vast programmer community, was the obvious choice. They envisioned Android "as the world's first Open Source handset solution

with built-in Google applications." Android GPS [Google Product Strategy]: Key strategic decisions around Open Source at 2 (July 26, 2005), Trial Ex. 1, *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 975 (N.D. Cal. 2012) (No. C 10-03561 WHA). An open source licensing model was critical to: (1) avoid the fragmentation resulting from closed and proprietary mobile platforms (like those offered by Microsoft and Symbian); (2) provide telecommunications companies and OEMs "a non-threatening solution for cross-vendor compatibility"; and (3) build a "community force around Google handset APIs and applications." *Id*.

Google engineers sought to use some of the Java APIs, but did not want to include the full set because of space and other design constraints. In addition, they wanted to add new APIs to support GPS, camera functions, and user preferences. To provide access to a subset of Java APIs, Google planned to develop a clean room implementation of the JVM and negotiate the first open source Java 2 Platform, Micro Edition JVM ("J2ME") license with Sun. After a promising start, negotiations broke down when Sun refused to budge on full interoperability with J2ME.

As a result, Google opted to build the Android operating system by emulating select Java API functionalities with independently written implementing code. The "clean room" process draws on Judge Learned Hand's originality corollary: "if by some magic a man who had never known it were to compose anew Keats's Ode on a Grecian Urn, he would be an 'author,' and, if he copyrighted it, others might not copy that poem,

though they might of course copy Keats's." *See Sheldon v. Metro-Goldwyn Pictures Corp.*, 81 F.2d 49, 54 (1936). The microcomputer industry developed around this principle. *See Sony Comput. Entertainment v. Connectix*, 203 F.3d 596, 599-608 (9th Cir. 2000); *Comput. Assocs. Int'l v. Altai, Inc.*, 982 F.2d 693, 700, 707-10 (2d Cir. 1992); *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510, 1526-28 (9th Cir. 1992); Network Effects and IP at 259-60. One team of programmers studies a program through legal means—such as by running the software to study its behavior, reviewing documentation, peeling semiconductor chips, decompiling object code—to determine its functional specifications. Those specifications are handed off to a second team of programmers with no knowledge of the implementing code of the target computer program. They independently develop a computer program with the same functionality as the target program. *See generally* P. Anthony Sammi, Christopher A. Lisy & Andrew Gish, Good Clean Fun: Using Clean Room Procedures in Intellectual Property Litigation, 25 INTELL. PROP. & TECH. L.J. 3 (2013).

If the Java programming language is analogized to musical language, each API implementation can be characterized as a record album featuring songs (methods). Java Standard Edition (SE) then functions like an electrical-mechanical juke box, containing API record albums from which programmers can choose particular songs by invoking declarations (song titles). The fact that another juke box uses those song titles (declarations) to invoke a known song (method) is purely *functional*: it does not copy a song (method), it merely identifies a known song (method).

Unlike the early software industry, which operated through trade secrecy and released only object code versions to the public, Sun published the Java API functional specifications (declarations) to encourage others to develop applications. Even with access to the functional specifications, emulating APIs can be difficult. During an arduous two year process, the Android team independently developed its own implementing code for 37 of the 166 Java API packages in Java SE and an independent JVM. In this way, the Android operating system emulated the functionality of known and tested APIs that fit the Android team's design. Android's use of the Java function labels (declarations) enabled millions of Java programmers to quickly master Android app development. Although Android apps were not fully interoperable with the Java SE platform, they were similar enough and better optimized to the constraints of Android mobile devices.

III. THE FEDERAL CIRCUIT'S DECISIONS CONFLICT WITH THIS COURT'S SEMINAL RULING IN *BAKER V. SELDEN*, MISINTERPRET CONGRESS'S CODIFICATION OF INTELLECTUAL PROPERTY LAW'S FUNDAMENTAL CHANNELING PRINCIPLE AND COPYRIGHT'S LIMITING DOCTRINES, AND UPEND WELL-REASONED AND LONG-STANDING JUDICIAL DECISIONS THAT HAVE SUPPORTED INNOVATION AND COMPETITION IN THE COMPUTER INDUSTRY

The unusual jurisdictional posture of the *Oracle v. Google* case required the Federal Circuit to apply Ninth Circuit law. *See Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1353 (Fed. Cir. 2014); Peter S. Menell, API Copyrightability Bleak House: Unraveling and Repairing the *Oracle v. Google* Jurisdictional Mess, 31 BERKELEY TECH. L.J. 1515, 1576–95 (2016). Although we question the Federal Circuit's fidelity to the Ninth Circuit's interpretation of copyright law, *see* Rise of the API Copyright Dead at 427-43, this Court confronts the interpretation of copyright protection for computer software as a question of national law. As suggested in Section I, we believe that this Court's seminal *Baker v. Selden* decision as later codified in the text and reflected in the legislative history of the Copyright Act provides the necessary blueprint. In a nutshell, the *functional requirements* of APIs, like the internal workings of other machines, stand outside the scope of copyright protection even as non-merged aspects of the implementing code for APIs are protectable under copyright law.

A. The Federal Circuit's Decisions Conflict with *Baker v. Selden*

The Federal Circuit's decisions directly conflict with *Baker v. Selden*, and the idea-expression and merger doctrines that it spawned. By affording Oracle exclusive rights to not just the implementing code for Java APIs but also the declarations necessary to call those methods, the Federal Circuit has protected the computer system's functionality through copyright law.

As Judge Alsup explained in the district court opinion,

> the rules of Java dictate the precise form of certain necessary lines of code called declarations, whose precise and necessary form explains why Android and Java must be identical when it comes to those particular lines of code. That is, since there is only one way to declare a given method functionality, everyone using that function must write that specific line of code in the same way.

Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974, 979 (N.D. Cal. 2012). Properly viewed, Sun's devising of a package (e.g., java.security) using a particular class name (e.g., ProtectionDomain) and method name (e.g., ClassLoader) to effectuate a machine that responds to particular inputs and produces particular outputs should be deemed to place the otherwise creative names and essential structure outside of copyright protection, thereby enabling others (in the absence of a utility patent covering this process or machine) to

emulate (and interoperate with) this machine so long as they write their own implementation. The Federal Circuit has undermined those core principles.

B. The Federal Circuit's Decisions Misconstrue the Copyright Act

Congress codified *Baker v. Selden*'s fundamental channeling principle in the text of the Copyright Act of 1976. 17 U.S.C. § 102(b). Beyond the statutory text, the legislative history states that "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*." H.R. REP. NO. 94-1476 at 57 (1976) (emphasis added).[4] CONTU clarifies that "one is always free to make a machine perform *any* conceivable process (in the absence of a patent) [so long as one does not] take another's program." CONTU REPORT at 20 (emphasis added).[5]

---

[4] The Federal Circuit omitted, without ellipses or explanation, the critical italicized completion to the quoted sentence. *See Oracle Am., Inc. v. Google Inc.*, 750 F.3d at 1367.

[5] Courts have treated the CONTU REPORT as legislative history for the Computer Software Act of 1980, amending the 1976 Act in accordance with CONTU's recommendations. *See Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 260-61 (5th Cir. 1988); *Apple Comput., Inc. v. Franklin Comput. Corp.*, 714 F.2d 1240, 1252 (3d Cir. 1983). *See generally* NIMMER ON COPYRIGHT § 8.08[A][3].

In accordance with the Copyright Act, Google was entitled to make a mobile device ("a machine") perform the same functions as a Java API package (a "conceivable process") with independently developed implementation code (i.e., not "another's program"). Each Java API package constituted a particular subsystem within a larger particular computing environment. Hence, Google was justified in selecting a set of Java API packages and implementing them with original code to create a new machine. *See* Rise of the API Copyright Dead at 433-52.

In support of its conclusions, the Federal Circuit also misconstrued this Court's observations in *Mazer v. Stein* that "[n]either the Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted." *Oracle Am., Inc. v. Google Inc.*, 750 F.3d at 1380. Contrary to the Federal Circuit's decision, this observation was made in relation to the overlap between copyright law and *design patent* law. *Mazer* states that "the Mechanical Patent Law and Copyright Laws are mutually exclusive" and that it is only with regard to *design patent* law that an overlap with copyright can occur. *Mazer v. Stein*, 347 U.S. 201, 215 n.33 (1954). *See* NIMMER ON COPYRIGHT § 2A.07[D][4][b].

Like copyright law, design patent protection is subservient to utility patent law with regard to functional elements. The design patent statute was based on England's design *copyright* statute. *See* Jason J. Du Mont & Mark D. Janis, The Origins of American Design Patent Protection, 88 INDIANA L.J. 837, 847, 854-73 (2013). The original bill proposed a "sole and

exclusive copy-right" for the proprietor of any "new and original design" for specified articles of manufacture. Ruggles Design Bill, S. 269, 26th Cong. § 1 (1841). Following Senator Ruggles' failed reelection bid, the design protection mantle was taken up by Patent Commissioner Henry Ellsworth, which resulted in an unfortunate labeling twist. In his 1841 Commissioner's Report to Congress, Commissioner Ellsworth called upon Congress to establish a design protection regime under his authority at the Patent Office. *See* Thomas B. Hudson, A Brief History of the Development of Design Patent Protection in the United States, 30 J. PAT. & TRADEMARK OFF. SOC'Y 380, 380-81 (1948). Although bringing protection for designs under Patent Office administration, the substance of the bill remained the same—protecting ornamental creativity as opposed to technological advances. The 1902 Design Patent Act, Act of May 9, 1902, ch. 783, § 4929, 32 Stat. 193 (1902), on which the 1952 Patent Act rests, makes this abundantly clear. Under the 1902 Act, "Any person who has invented any new, original, and *ornamental* design for an article of manufacture" (emphasis added), may apply for a design patent.

Hence, the logic of *Baker v. Selden* applies equally to design patents as it does to copyrights. To the extent that an expressive feature of a design otherwise eligible for protection under either the Copyright Act *or the design patent provisions of the Patent Act* is merged or inextricably intertwined with a functional element, it can only be protected under utility patent law.

Accordingly, the Federal Circuit's reliance on *Mazer* was misplaced.

C. The Federal Circuit's Decisions Revive and Exacerbate Circuit Splits on Copyrightability, Merger, and Fair Use

Copyright protection for computer software got off to an inauspicious start. *See* Rise of the API Copyright Dead at 322-26. In a case involving blatant and cavalier piracy of entire computer programs, the Third Circuit went overboard in addressing the defendant's interoperability argument, stating in dicta that "total compatibility with independently developed application programs . . . is a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged." *Apple Comput., Inc. v. Franklin Comput. Corp.*, 714 F.2d 1240 (3d Cir. 1983). Building on that decision, the Third Circuit ruled, in distinguishing protectable expression from unprotectable ideas, that

> *the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea.* Where there are many means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea.

Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1236 (3d Cir. 1986) (emphasis in

original; citations omitted). In applying this rule, the court defined the idea as "the efficient management of a dental laboratory," for which countless ways of expressing the idea would be possible. *Id*.

These decisions were roundly criticized. *See*, *e.g.*, NIMMER ON COPYRIGHT § 13.03[F], at 13-62.34, at 13–4 (1991) (explaining that "[t]he crucial flaw in [*Whelan*'s] reasoning is that it assumes that only one 'idea,' in copyright law terms, underlies any computer program, and that once a separable idea can be identified, everything else must be expression"); Donald S. Chisum, et al., LaST Frontier Conference on Copyright Protection of Computer Software, 30 JURIMETRICS J. 15, 20-21 (1989); Peter S. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 STAN. L. REV. 1045, 1074 (1989). As this scholarship emphasized, viewing the idea-expression dichotomy at such a high level of abstraction produced an overbroad scope of copyright protection because it resulted in all implementations of the idea garnering protection. Furthermore, the Third Circuit's misreading of merger analysis and the idea-expression doctrine implicitly allowed copyright protection of procedures, processes, systems, and methods of operation that are expressly excluded under § 102(b). Drawing on this scholarship, other circuits developed alternative approaches to the scope of copyright protection that better comported

with *Baker v. Selden* and related fundamental limiting doctrines.[6]

Thus, after a rocky start, the regional circuit courts of appeals implemented a balanced framework for both protecting computer software against piracy and interpreting the idea-expression doctrine to ensure that copyright law excludes functional features of computer technology. Although the early overbroad Third Circuit cases remained on the books, software copyright law achieved stability, clarity, and sound reasoning. *See Sony Comput. Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000) (reinforcing and extending *Sega*); *Lexmark Int'l, Inc. v. Static Control*

---

[6] *See Plains Cotton Coop. Assoc. v. Goodpasture Comput. Serv., Inc.*, 807 F.2d 1256, 1262 (5th Cir. 1987) (declining to follow *Whelan*); *Comput. Assocs. Int'l v. Altai, Inc.*, 982 F.2d 693, 705-06 (2d Cir. 1992) (rejecting dictum in *Apple v. Franklin* and the *Whelan Associates* analytical framework, and applying an abstraction-filtration-comparison framework based on Judge Learned Hand's classic mode of analysis in *Nichols v. Universal Pictures Corp.*, 45 F.2d 119 (2d Cir. 1930)); *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1525 (9th Cir. 1992) (rejecting *Whelan*); *Gates Rubber v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 834, 841 (10th Cir. 1993) (endorsing *Altai*); *Apple Computer, Inc. v. Microsoft Corp.*, 799 F. Supp. 1006 (N.D. Cal. 1992), *aff'd in part, rev'd in part*, 35 F.3d 1435, 1445 (9th Cir. 1994) (endorsing *Altai*); *Eng'g Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1341-43 (5th Cir. 1994) (endorsing *Altai*); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 813 (1st Cir. 1995) (holding that a menu command hierarchy that functions as a programming language is an uncopyrightable method of operation), *aff'd by an equally divided Court*, 516 U.S. 233 (1996); *MiTek Holdings, Inc. v. ARCE Engineering Co.*, 89 F.3d 1548, 1559 (11th Cir. 1996) (endorsing *Altai*); *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1375 (10th Cir. 1997) (endorsing *Altai*).

Components, Inc., 387 F.3d 522, 534-37 (6th Cir. 2004) (reinforcing this evolution); *see generally* Peter S. Menell, An Epitaph for Traditional Copyright Protection of Network Features of Computer Software, 43 ANTITRUST BULL. 651, 707-08 (1998).

By the mid to late 1990s, the software engineering community came to view high-level functions, labeling conventions, and the functional specifications of APIs as unprotectable under copyright law. *See* Brian Profitt, The Impact of Oracle's Defense of API Copyrights, ITWORLD (Aug. 23, 2011) (observing that "[h]istorically, APIs have been regarded as not falling under copyright—the reasoning being that APIs are not creative implementations but rather statements of fact," but also noting the issue had been clouded by the distinction between "open" and "closed" APIs). The Federal Circuit's *Oracle v. Google* decisions revived and exacerbated the long dormant circuit splits relating to copyrightability of particular elements of computer software, copyright infringement analysis, and the application of the Copyright Act's fair use doctrine. *See* Michael Hussey, Copyright Captures APIs: A New Caution for Developers, TECHCRUNCH (Nov. 3, 2015).

D. Reversing the Federal Circuit's *Oracle v. Google* Decisions Maintains the Coherence of the Intellectual Property System and Will Restore Peace and Clarity to the Computer Software Industry

Resolving this appeal through the clear and logical principles underlying *Baker v. Selden* maintains the coherence of the intellectual property system. The Supreme Court has reinforced these principles in its modern jurisprudence. *See Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 146 (1989) (observing that Art. I, § 8, cl. 8 of U.S. Constitution "reflects balance between the need to encourage innovation and the avoidance of monopolies which stifle competition without any concomitant advance in the 'Progress of Science and useful Arts' "); *TrafFix Devices, Inc. v. Marketing Displays, Inc.*, 532 U.S. 23, 29-30 (2001) ("Where the expired patent claimed the features in question, one who seeks to establish trade dress protection must carry the heavy burden of showing that the feature is not functional, for instance by showing that it is merely an ornamental, incidental, or arbitrary aspect of the device."); *Wal-Mart Stores, Inc. v. Samara Brothers, Inc.*, 529 U.S. 205, 213 (2000) (cautioning against protecting functional features through trademark law); *cf. Dastar Corp. v. Twentieth Century Fox Film Corp.*, 539 U.S. 23, 34 (2003) (warning that "allowing a cause of action under [Lanham Act] § 43(a) for reverse passing off would create a species of mutant copyright law that limits the public's 'federal right to "copy and to use" ' expired copyrights").

Following the extensive litigation over copyright protection for APIs from the early 1980s through the mid-1990s, the regional circuit courts of appeals achieved a balanced approach that afforded protection for API implementations without hindering competition in the computer industry. The computer software industry embraced that balance and was able to function efficiently for more than a decade. The Federal Circuit's *Oracle v. Google* decisions cast a dark cloud over standard industry practice. Restoring the clarity that prevailed prior to the *Oracle v. Google* decisions will promote innovation in an industry driven by complex, time-sensitive decisions.

Reversing the *Oracle v. Google* decisions will not deprive API developers of adequate intellectual property protection. They retain protection against companies that copy their implementation. Moreover, by employing technological protection measures or distributing software products solely in object code form, software developers can slow development of interoperable products. Reverse engineering computer programs is laborious, slow, and expensive. Even when the functional specifications are publicly disclosed, as was the case with the Java APIs, re-implementing that functionality in a clean room is often costly and time-consuming. And API developers can seek utility patents on technological innovations.

IV. GOOGLE INDEPENDENTLY IMPLEMENTED THE JAVA API FUNCTIONAL SPECIFICATIONS AND THEREFORE DID NOT INFRINGE ORACLE'S COPYRIGHTS

The record in this case establishes that after the Google-Sun negotiations over licensing the API implementations reached an impasse, Google undertook the burdensome and costly effort to implement the APIs independently. Although Oracle initially questioned the independence of those implementations, it conceded that Google's implementation did not infringe the Java API implementations other than through the copying of Java declarations and their structure, sequence, and organization. Inclusion of declarations was necessary to achieve the particular API functionality. Unless Google used the precise declarations, Android could not emulate the functionality of the Java APIs.

Like the labels and columns of Selden's accounting forms in *Baker v. Selden*, Java API declarations are not protected by copyright because they are essential to invoke, access, unlock, or operate a particular system or machine (specific Java API functions) even though the particular implementation code for those functional specifications is copyright-protected. To protect API declarations through copyright law would afford Oracle patent-like protection over particular machine functions.

To make this point concrete, suppose that Sega had written its lockout code not as a peculiar sequence

of data, *Sega*, 955 F.2d at 1516, but rather as an original haiku. Even though that haiku could be protected if distributed as poetry, it would be barred from copyright protection as lockout code. That is the reason for the Ninth Circuit's unmistakable statement that the "functional requirements for compatibility with the Genesis [video game console are] aspects of Sega's programs that are not protected by copyright. 17 U.S.C. § 102(b)." *Id*. at 1522.[7] As essential "gears and levers" for particular digital machines, the Java API declarations are not protectable under copyright law due to the overarching channeling principles reflected in *Baker v. Selden* and § 102(b) of the Copyright Act.

It is for that reason that it is irrelevant that the Java APIs might be highly creative. So are haikus. But if used to operate functional devices, their function merges with the expression and copyright cannot limit the functional use. Technological creativity is often among the most difficult, imaginative, and praiseworthy forms of creativity. Yet the overarching intellectual property system would be undermined if an inventor of a better analog cash register could bar competition

---

[7] That case simultaneously rejected Sega's attempt to invoke trademark laws to protect the lockout code that it voluntarily adopted, namely "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD." *Id*. at 1515. Having used that formulation, Sega could not complain when third parties similarly employed it to gain the necessary access. *Id*. at 1528-30. As discussed above in § III(D), the coherence of the intellectual property system disallows overly aggressive plaintiffs from deploying "copyright," "trademark," or any other label for the improper purpose of excluding competitors from access to legitimate domains in order to achieve program compatibility.

for life of the inventor plus 70 years by copyrighting the configuration of gears and levers as sculptural works or if a computer programmer who devised a better digital cash register (e.g., a spreadsheet) could protect its invention for her or his life plus 70 years through copyright protection for the functional specifications (declarations).

The Federal Circuit confused the infringement analysis by accepting Oracle's characterization of declarations as "declaring code." But declarations are not code but rather textual labels that specify properties of an identifier. A declaration

> declares what a word (identifier) 'means.' Declarations are most commonly used for functions, variables, constants, and classes, but can also be used for other entities such as enumerations and type definitions. Beyond the name (the identifier itself) and the kind of entity (function, variable, etc.), declarations typically specify the data type (for variables and constants), or the type signature (for functions); types may also include dimensions, such as for arrays. A declaration is used to announce the existence of the entity to the compiler.

Declaration, WIKIPEDIA.

Nor do the API declarations attract protection on the ground that their structure, sequence, and organization is protectable. Combinations of methods in a machine are no more copyrightable than the configuration of tools in a Swiss Army knife. Moreover, these labels are essential to accessing and operating a particular computing machine or method. The implementing

code is protectable, but not the declarations or structure of declarations necessary to operate a particular machine, method, or other functional component of a machine.

V. THE FAIR USE ISSUE IS MOOT

In view of the foregoing establishing that Google did not infringe Oracle's Java APIs copyrights, there is no reason for the Court to reach the fair use question. *Cf.* Rise of the API Copyright Dead at 471-73 (explaining that reliance on the inherently costly, time-consuming, and unpredictable fair use doctrine is ill-suited for resolving API copyright disputes).

CONCLUSION

For the reasons set forth above, the Court should reverse the Federal Circuit's 2014 and 2018 *Oracle v. Google* decisions and rule that Google has not infringed Oracle's Java API copyrights.

Respectfully submitted,

PETER S. MENELL
KORET PROFESSOR OF LAW
Counsel of Record
UNIVERSITY OF CALIFORNIA,
AT BERKELEY SCHOOL OF LAW
225 Bancroft Way
Berkeley, CA 94720-7200
(510) 642-5489
pmenell@law.berkeley.edu

January 10, 2020