

No. 14-410

---

---

IN THE  
**Supreme Court of the United States**

---

GOOGLE INC.,

*Petitioner,*

*v.*

ORACLE AMERICA, INC.,

*Respondent.*

---

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

---

---

**BRIEF OF *AMICI CURIAE* INTELLECTUAL  
PROPERTY PROFESSORS IN SUPPORT OF  
GRANT OF PETITION**

---

---

PAMELA SAMUELSON  
*Counsel of Record*  
UNIVERSITY OF CALIFORNIA, BERKELEY,  
SCHOOL OF LAW  
434 Boalt Hall  
Berkeley, California 94720  
(510) 642-6775  
pam@law.berkeley.edu

*Counsel of Record for Amici Curiae  
Intellectual Property Professors*

---

---

256245



COUNSEL PRESS

(800) 274-3321 • (800) 359-6859

**TABLE OF CONTENTS**

	<i>Page</i>
TABLE OF AUTHORITIES.....	ii
INTEREST OF <i>AMICI CURIAE</i> .....	1
SUMMARY OF ARGUMENT.....	2
ARGUMENT	
I. Copyright Protection Does Not Extend to Procedures, Processes, Systems, Methods of Operation, and Other Useful Arts Embodied in Computer Programs Under Section 102(b). .....	5
II. The Judicially Developed Doctrine on the Structure, Sequence and Organization of Computer Programs Should Not Be Invoked in a Way That Negates Congress’s Clear Intent in Section 102(b). .....	13
III. When Computer Program Interfaces Constrain the Design Choices of Subsequent Programmers, the Merger Doctrine Precludes Copyright Protection for the Interface Design. ....	18
CONCLUSION .....	24
APPENDIX: List of Signatories.....	1a

## TABLE OF AUTHORITIES

*Page(s)*

### Cases

<i>Apple Computer, Inc. v. Franklin Computer Corp.</i> , 714 F.2d 1240 (3d Cir. 1983) .....	21, 22, 23
<i>Atari Games Corp. v. Nintendo of America Inc.</i> , 975 F.2d 832 (Fed. Cir. 1992) .....	12, 20
<i>Baker v. Selden</i> , 101 U.S. 99 (1879) .....	8, 9, 21
<i>Bateman v. Mnemonics, Inc.</i> , 79 F.3d 1532 (11th Cir. 1996) .....	8, 18
<i>Brown Instrument Co. v. Warner</i> , 161 F.2d 910 (D.C. Cir. 1947) .....	11
<i>Computer Assocs. Int’l, Inc. v. Altai, Inc.</i> , 982 F.2d 693 (2d Cir. 1992) .....	7, 8, 14, 15, 18, 19, 20
<i>Gates Rubber Co. v Bando Chem. Indus., Ltd.</i> , 9 F.3d 823 (10th Cir. 1993) .....	8, 16
<i>Hutchins v. Zoll Med. Corp.</i> , 492 F.3d 1377 (Fed. Cir. 2007) .....	16, 18
<i>Lexmark Int’l, Inc. v. Static Control Components, Inc.</i> , 387 F.3d 522 (6th Cir. 2004) .....	8, 20
<i>Lotus Dev. Corp. v. Borland Int’l, Inc.</i> , 49 F.3d 807 (1st Cir. 1995), <i>aff’d by an equally</i>	

<i>divided Court</i> , 516 U.S. 233 (1996) .....	8, 16, 17
<i>Lotus Dev. Corp. v. Paperback Software Int'l</i> , 740 F. Supp. 37, 67 (D. Mass. 1990).....	14
<i>M. Kramer Mfg. Co. v. Andrews</i> , 783 F.2d 421, 436 (4th Cir. 1986) .....	20
<i>Mitel, Inc. v. Iqtel, Inc.</i> , 124 F.3d 1366 (10th Cir. 1997) .....	18
<i>Oracle Am., Inc. v. Google Inc.</i> , 872 F. Supp. 2d 974 (N.D. Cal. 2012) .....	7, 11, 12, 14, 19, 23
<i>Oracle Am., Inc. v. Google Inc.</i> , 750 F.3d 1339 (Fed. Cir. 2014) .....	6, 13, 19
<i>Sega Enters., Ltd. v. Accolade, Inc.</i> , 977 F.2d 1510 (9th Cir. 1992) .....	6, 7, 15, 16, 18, 21, 22
<i>Sony Computer Entm't, Inc. v. Connectix, Corp.</i> , 203 F.3d 596 (9th Cir. 2000) .....	7, 22
<i>Taylor Instrument Co. v. Fawley-Brost Co.</i> , 139 F.2d 98 (7th Cir. 1943) .....	10, 11
<i>Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.</i> , 797 F.2d 1222 (3rd Cir. 1986) .....	14
 <b>Statutes and Other Authorities</b>	
17 U.S.C. § 102(b) .....	<i>passim</i>

Christina Bohannon, <i>Reclaiming Copyright</i> , 23 Cardozo Arts & Ent. L.J. 567 (2006) .....	17
Michael W. Carroll, <i>One For All: The Problem of Uniformity Cost in Intellectual Property Law</i> , 55 Am. U. L. Rev. 845 (2006) .....	17
Thomas F. Cotter, <i>The Procompetitive Interest in Intellectual Property Law</i> , 48 Wm. & Mary L. Rev. 483 (2006) .....	17
Stacey L. Dogan & Joseph P. Liu, <i>Copyright Law and Subject Matter Specificity: The Case of Computer Software</i> , 61 N.Y.U. Ann. Surv. Am. L. 203 (2005) .....	17
Herbert Hovenkamp, <i>Response: Markets in IP and Antitrust</i> , 100 Geo. L.J. 2133 (2012) .....	17
H.R. Rep. No. 94-1476 (1976) .....	5-6
Dennis S. Karjala, <i>A Coherent Theory for the Copyright Protection of Computer Software and Recent Judicial Interpretations</i> , 66 U. Cin. L. Rev. 53 (1997) .....	17
Peter Lee, <i>The Evolution of Intellectual Infrastructure</i> , 83 Wash. L. Rev. 39 (2008) .....	17
Aaron K. Perzanowski, <i>Rethinking Anticircumvention's Interoperability Policy</i> , 42 U.C. Davis L. Rev. 1549 (2009) .....	17

- Randal C. Picker, *Competition and Privacy in Web 2.0 and the Cloud*, 103 Nw. U. L. Rev. Colloquy 1 (July 28, 2008), <http://www.law.northwestern.edu/lawreview/colloquy/2008/25/> ..... 17-18
- J.H. Reichman & Jonathan A. Franklin, *Privately Legislated Intellectual Property Rights: Reconciling Freedom of Contract with Public Good Uses of Information*, 147 U. Pa. L. Rev. 875 (1999) ..... 18
- S. Rep. No. 94-473 (1975) ..... 5-6
- Pamela Samuelson, *Why Copyright Law Excludes Systems and Processes from the Scope of Its Protection*, 85 Tex. L. Rev. 1921 (2007) ..... 9, 10
- Philip J. Weiser, *The Internet, Innovation, and Intellectual Property Policy*, 103 Colum. L. Rev. 534 (2003) ..... 18

**INTEREST OF *AMICI CURIAE*<sup>1</sup>**

*Amici curiae* are individuals who teach and write about intellectual property law at accredited law schools in the United States. A list of *Amici* appears in Appendix A. *Amici* respectfully submit this brief to express our views and our concerns. Our sole interest in this case is with respect to a number of traditional principles of copyright law that we, as instructors and commentators on intellectual property law, believe should be considered in determining the proper scope of copyright protection for certain elements of computer programs. We write to urge this Court to reverse the Federal Circuit ruling that the Java application program interfaces (APIs) at issue in this case are protectable expression under U.S. copyright law.

---

<sup>1</sup> No counsel for a party authored this brief in whole or in part, and no such counsel or party made a monetary contribution intended to fund the preparation or submission of this brief. No person other than *Amici* or their counsel made a monetary contribution to its preparation or submission. Petitioner has consented to the filing of this brief and has filed a letter with the Court granting blanket consent. Respondent has granted written consent to the filing of this brief. Parties received at least ten days' notice of *Amici's* intention to file this brief.

## SUMMARY OF ARGUMENT

In ruling that the Java APIs at issue in this case are protectable expression as a matter of U.S. copyright law, the Federal Circuit has incorrectly applied binding Ninth Circuit precedents and issued an opinion that splits from rulings by the First, Second, Sixth, Ninth, Tenth and Eleventh Circuit Courts of Appeals. A reversal is warranted because of three foundational errors in the Federal Circuit's analysis of software copyright law in the *Oracle* case.

First, the Federal Circuit took an unduly narrow view of 17 U.S.C. § 102(b), which provides that “[i]n no case does copyright protection . . . 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 a protected work. (Emphasis added.) The legislative history of the Copyright Act of 1976 reveals that Congress was well aware that computer programs instantiate numerous types of unprotectable methods of operation and processes. Indeed, Congress put the Section 102(b) exclusions in the statute with the specific intent of ensuring that copyright protection for programs would not be interpreted too broadly. The Federal Circuit *Oracle* ruling contravenes a traditional canon of statutory construction to give all words in a statute appropriate meaning.

After a full trial on the merits and consideration of numerous briefs, the District Court determined that the command structures of the Java APIs at



issue in this case were unprotectable methods of operation under Section 102(b), in keeping with controlling Ninth Circuit decisions. The Federal Circuit ignored the findings of fact on which that ruling was based, misconstrued the Ninth Circuit precedents that had rightly held that interfaces are unprotectable under Section 102(b), and erroneously insisted that interoperability can only be taken into account in considering whether a defendant has made fair use of another program's APIs.

Second, the Federal Circuit was mistaken in its overbroad assessment about the degree to which the "structure, sequence and organization" (SSO) of computer programs are protectable by copyright law. During the 1980s, some courts found that program SSO could be protected by copyright law. Since then, courts and commentators have recognized that the SSO concept is too imprecise and misleading to be useful in software copyright cases. The SSO concept does not help courts make appropriate distinctions between protectable and unprotectable structural elements of programs. Procedures, processes, systems, and methods of operation, almost by definition, contribute to the SSO of programs that embody them. However, this does not make those elements protectable by copyright. When the design of program structures, such as APIs, is inherently functional and aimed at achieving technical goals of efficiency, they do not qualify as protectable expression under U.S. law.

Third, the Federal Circuit has a mistaken understanding of the merger doctrine as applied in computer program copyright cases. The Federal Circuit asserts that as long as Oracle's engineers

exhibited creativity in the initial design of Java APIs and those engineers were not constrained in their choices about how to construct the APIs, the APIs are *ab initio* copyright-protectable expression. However, caselaw from numerous circuits recognizes that when the design choices of subsequent programmers are constrained by the interface designs embodied in earlier programs, the merger doctrine applies so that programmers can reuse elements necessary to achieve interoperability. All that subsequent programmers must do is to reimplement interface elements in independently created code. In its findings of fact, the District Court found that Google had done exactly this. APIs are fundamental building blocks of program development.

This Court should reject the Federal Circuit's erroneous interpretation of the merger doctrine as applied to computer programs. If it does not, the result will likely be a surge in litigation over the protectability of APIs, even though this issue had seemed to be resolved by several Circuit Court of Appeals rulings going back to 1992. Since then, innovators have relied on the understanding that the interface elements embodied in APIs are unprotectable, and competition and innovation in the U.S. software industry has thrived in part because of rulings that support this understanding. The decision below has opened up a new period of uncertainty about the copyrightability of APIs. Unless reversed, this is likely to have a chilling effect on the willingness of software startup firms and open source developers to create innovative programs that can be executed on incumbent firm platforms.

Competition and innovation in the software industry will suffer unless this Court resolves the circuit split on this issue and repudiates the Federal Circuit's ruling.

## ARGUMENT

### **I. Copyright Protection Does Not Extend to Procedures, Processes, Systems, Methods of Operation, and Other Useful Arts Embodied in Computer Programs Under Section 102(b).**

Section 102(b) of the Copyright Act of 1976 states plainly that “[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle or discovery, regardless of the form in which it is . . . embodied in such work.” 17 U.S.C. § 102(b). The House and Senate Reports explain why this provision was included in the statute:

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 (1976); S. Rep. No. 94-473, at 54 (1975).

A glaring error in the Federal Circuit’s *Oracle* ruling is its failure to heed Ninth Circuit and other circuit caselaw interpreting this provision as applied to program APIs. It seemed to regard Section 102(b) as merely a restatement of the distinction between ideas and expression. *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1353 (Fed. Cir. 2014). Although it recognized that it was obliged to follow Ninth Circuit precedent, *id.* at 1352, the Federal Circuit misconstrued the most relevant Ninth Circuit rulings.

First, the Federal Circuit failed to acknowledge the Ninth Circuit’s ruling that “the functional requirements for [achieving] compatibility” are aspects of computer programs that “are not protected by copyright” because these interface procedures are excluded from protection under 17 U.S.C. § 102(b). *Sega Enters., Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1522 (9th Cir. 1992).<sup>2</sup> To attain a lawful monopoly over functional principles underlying a work, such as program interface procedures, the Ninth Circuit has said that “the creator of the work must satisfy the more stringent standards imposed by the patent laws.” *Id.* at 1526. In *Sega*, the court also held that copying exact lines of code was non-infringing when

---

<sup>2</sup> The unprotectability of the *Sega* interface procedures was not a “tangential[]” aspect of the *Sega* ruling, as the Federal Circuit thought. *Oracle*, 750 F.3d at 1369. The fair use ruling was grounded on the interface procedures *Accolade* reverse-engineered being unprotectable elements of the *Sega* programs.

that code was essential to achieving interoperability. *Id.* at 1522-23, 1524. The Ninth Circuit reaffirmed that Section 102(b) makes the functional requirements for achieving compatibility unprotectable by copyright law in *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596, 603 (9th Cir. 2000).

The District Court followed these Ninth Circuit precedents in ruling that the Java APIs at issue in this case were unprotectable methods or systems under Section 102(b). *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 977, 994-97. It observed that “the rules of Java dictate the precise form of 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.” *Id.* at 979. Echoing the Ninth Circuit in *Sega*, the District Court indicated that the Java API command structure might be eligible for patent, but not for copyright, protection. *Id.* at 999-1000

The Ninth Circuit in *Sega* was influenced by the landmark software copyright decision *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), which also ruled that the functional requirements for achieving interoperability were not protectable by copyright law. *Altai* at 709-10. The Second Circuit in *Altai* recognized that copyrightability is limited for computer programs, which are “process oriented text[s]” that “hover . . . closely to the elusive boundary line described in § 102(b).” *Id.* at 704. The methods and processes explained or embodied in

such works lie outside the scope of copyright protection available to them. *Id.*

The *Altai* and *Sega* rulings have been followed by the Sixth, Tenth, and Eleventh circuits. *See, e.g., Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 534-36 (6th Cir. 2004) (program compatibility component held unprotectable under the merger doctrine and Section 102(b)); *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 842-45 (10th Cir. 1993) (reversing lower court ruling for failure to exclude unprotectable elements of programs); *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532, 1546-47 (11th Cir. 1996) (error not to instruct jury as to compatibility defense). Giving a broader interpretation of Section 102(b), the First Circuit held that methods of operation embodied in programs are excluded from protection in *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).

The Federal Circuit's reasoning in *Oracle* calls all of these rulings into question and represents a clear circuit split.

Second, the Federal Circuit erred in rejecting the District Court's sound application of Section 102(b) and precedent stretching back to 1879. *Amici* agree with the court in *Altai* that "[t]he doctrinal starting point in analyses of utilitarian works [such as computer programs] is the seminal case" of *Baker v. Selden*, 101 U.S. 99 (1879). *Altai*, 982 F.2d at 704. *Baker* is the origin of the rule that systems, methods, and other useful arts embodied in copyrighted works are not within copyright's scope of protection. The

issue in *Baker* was “whether the exclusive property in a system of book-keeping can be claimed, under the law of copyright, by means of a book in that system is explained[.]” *Baker*, 101 U.S. at 101. Selden claimed that Baker infringed by copying the selection and arrangement of columns and headings from the forms Selden designed to illustrate how to implement his novel bookkeeping system. *See id.*

This Court ruled that Selden’s copyright extended to his explanation of the bookkeeping system, but not to the system itself or to the forms that implemented it. *See Baker*, 101 U.S. at 100-02. The bookkeeping system was a “useful art” that might be eligible for patent protection, but not for copyright. The Court explained:

To give to the author of the book an exclusive property in the [useful] 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 [a useful] 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. This was relevant in *Baker* because the preface to Selden’s book revealed that he had sought (and apparently not obtained) a patent on the very same bookkeeping system over which he was suing

Baker in the copyright lawsuit. See Pamela Samuelson, *Why Copyright Law Excludes Systems and Processes from the Scope of Its Protection*, 85 Tex. L. Rev. 1921, 1929, 1933 (2007).

Between 1880 and 1976, dozens of cases followed *Baker*, extended its analysis to a wide variety of subject matters beyond bookkeeping methods and systems, and offered additional insights about why such things as systems, methods, processes, and procedures should be excluded from the scope of copyright. See Samuelson, *supra*, at 1936-44 (discussing the post-*Baker* caselaw). Section 102(b) was intended to codify the exclusion of processes, systems, and methods of operation from the scope of copyright in keeping with the post-*Baker* caselaw. See *id.* at 1944-61.

Excluding methods and processes from the scope of copyright helps to preserve the distinction between patent and copyright protection, as the Seventh Circuit explained in *Taylor Instrument Co. v. Fawley-Brost Co.*, 139 F.2d 98 (7th Cir. 1943). Taylor had registered claims of copyright in several hundred charts for use in connection with its temperature recording machines. It charged Fawley-Brost with infringing eighteen of these copyrights by making and selling charts that were virtually identical to Taylor's charts and hence interoperable with Taylor's machines. See *id.* at 99. The Seventh Circuit, invoking *Baker*, ruled against this claim because the charts, as integral components of



temperature recording systems, were not protectable by copyright law. *See id.* at 99-101.<sup>3</sup>

The court in *Taylor* perceived Congress to have provided “two separate and distinct fields of protection, the copyright and the patent,” and to have placed writings of authors in the former and inventive useful arts in the latter. *Id.* at 99. “[I]t must be in one or the other; it cannot be found in both.” *Id.* The court took into account that many patents had issued for temperature recording machines and charts for use in connection with them. Its examination of Taylor’s recording devices and charts left “no room for doubt but that the latter is a mechanical element of the instrument of which it is an integral part.” *Id.* at 100.

The court in *Taylor* observed that “the chart neither teaches nor explains the use of the art. It is an essential element of the machine; it is the art itself.” *Id.* Upholding Taylor’s claim would “produce [an] intolerable situation” because Taylor could “extend indefinitely the fifty-six years of protection afforded by the copyright laws” by changing the configuration of its machines and thwart competition by firms such as Fawley-Brost. *Id.* at 101.

The District Court in this case cited to *Baker* in expressing a similar concern that Oracle might be seeking through its copyright claim against Google to obtain “an exclusive right to a functional system,

---

<sup>3</sup> Soon thereafter, the Register of Copyrights denied an application to register copyrights in similar charts, and the D.C. Circuit affirmed this rejection relying on *Baker* and *Taylor*. *See Brown Instrument Co. v. Warner*, 161 F.2d 910, 910-11 (D.C. Cir. 1947).

process, or method of operation that belongs in the realm of patents, not copyrights.” *Oracle*, 872 F. Supp. 2d at 984. The court noted that “[b]oth Oracle and Sun have applied for and received patents that claim aspects of the Java API.” *Id.* at 996.

The Federal Circuit in *Oracle* is simply wrong in its interpretation of Section 102(b) and in its view that patent and copyright have overlapping roles in protecting computer program innovations. Functional methods and processes embodied in computer programs may be eligible for patent protection, but not for copyright. The Federal Circuit itself previously acknowledged the separate roles of utility patents and copyrights in the protection of computer programs. In *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992), after quoting the pertinent parts of Section 102(b) the court observed:

In conformance with the standards of patent law, title 35 provides protection for the process or method performed by a computer in accordance with a program. Thus, *patent and copyright laws protect distinct aspects of a computer program*. Title 35 protects the process or method performed by a computer program; title 17 protects the expression of that process or method. While title 35 protects any novel, nonobvious, and useful process, title 17 can protect a multitude of expressions that implement that process.

*Id.* at 839 (emphasis added) (citations omitted).

The District Court's interpretation of Section 102(b) was sound, for it heeded two key statutory principles: first, that copyright protection is available to protect expressive aspects of program code, and second, that procedures, processes, systems, and methods of operations embodied in programs are unprotectable by copyright law.

In overruling the District Court, the Federal Circuit, in effect, read the procedure, process, system, and method exclusions out of the statute. It is a basic canon of statutory interpretation that courts must endeavor to give all words in a statute appropriate meaning. If Congress has decided that computer programs are copyrightable, but methods of operation, systems, procedures, and processes embodied in programs are not, then it is incumbent on courts to determine which methods and processes in programs are beyond the scope of copyright protection.

## **II. The Judicially Developed Doctrine on the Structure, Sequence and Organization of Computer Programs Should Not Be Invoked in a Way That Negates Congress's Clear Intent in Section 102(b).**

The Federal Circuit ruled that Google infringed copyright in the "structure, sequence and organization" (SSO) of the Java APIs at issue in this case because Google could have organized API packages in a number of different ways. *Oracle*, 750 F.3d at 1367. In doing so, it departed from both the best-reasoned appellate cases considering this point and the Ninth Circuit's adoption of that reasoning. This merits review.

The idea that program SSO is protectable expression as long as there is more than one way to accomplish the objective derives from *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1236 (3rd Cir. 1986); accord *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 67 (D. Mass. 1990). Like the Federal Circuit, *Whelan* treated Section 102(b) as nothing more than a restatement of the idea/expression distinction. *Whelan*, 797 F.2d at 1234.

The SSO concept was discredited in the Second Circuit's *Altai* decision. *Altai*, 982 F.2d at 705-06. In the more than two decades since *Altai*, courts have largely moved away from conceiving of SSO as protectable expression in programs because they have realized that it fails to provide a workable framework within which to distinguish protectable and unprotectable structural aspects of programs. Since *Altai*, the trend in the caselaw "has been driven by fidelity to Section 102(b) and recognition of the danger of conferring a monopoly by copyright over what Congress expressly warned should be conferred only by patent." *Oracle*, 872 F. Supp. 2d at 996.

The Second Circuit decision in *Altai* emphasized that the "essentially utilitarian nature of computer programs" makes it difficult to separate protectable and unprotectable structural elements in programs. *Altai*, 982 F.2d at 704. As an alternative to focusing on SSO, the court announced a new "abstraction, filtration, and comparison" test for software copyright infringement. *Id.* at 706. Among the structural elements of programs that must be filtered out before assessing infringement are

efficient design elements, elements constrained by external factors, and standard programming techniques. *Id.* at 707-10.

The Second Circuit in *Altai* affirmed the lower court's finding that the similarities between Computer Associates' (CA's) and Altai's programs were "dictated by the functional demands" of the programs at issue or were otherwise in the public domain. *Id.* at 714. Altai needed to have similar lists of services and parameter lists because both CA's and Altai's scheduling programs were designed to provide the same services and conform to the interface procedures necessary to interoperate with IBM system programs. *See id.* at 715.

The *Altai* court was quite explicit that elements of programs "dictated by external factors" such as "compatibility requirements of other programs with which a program is designed to operate in conjunction" lie outside the scope of protection that copyright provides to programs. *Id.* at 709-10. Such structural similarities must be filtered out before courts can determine whether a defendant infringed copyright. *See id.*

The Ninth Circuit followed *Altai*'s lead in holding that interface procedures necessary for achieving interoperability among programs were functional elements of programs that copyright did not protect under Section 102(b). *See Sega*, 977 F.2d at 1522. In *Sega*, the court cited approvingly to *Altai* for the proposition that computer programs "contain many logical, structural, and visual display elements that are dictated by the function to be performed, by considerations of efficiency, or by external factors

such as compatibility requirements and industry demands.” *Id.* at 1524. Accolade had to reverse engineer Sega programs to get access to Sega interface procedures so that it could reimplement those procedures in independently written code. *See id.* at 1522.

Interface procedures are not the only structural design elements that are beyond the scope of copyright under Section 102(b). Algorithms, like interface procedures, are unquestionably elements of program SSO. Yet they too are beyond the scope of copyright protection as unprotectable procedures and processes. *See Gates Rubber Co. v Bando Chem. Indus., Ltd.*, 9 F.3d 823, 844-45 (10th Cir. 1993).

The structure and sequence of functional tasks that programs are designed to perform are unprotectable processes under Section 102(b), as the Federal Circuit recognized in *Hutchins v. Zoll Medical Corp.*, 492 F.3d 1377 (Fed. Cir. 2007). In *Hutchins*, the Federal Circuit held that copyright protection was unavailable to the “technologic method of treating victims by using CPR and instructing how to use CPR.” *Id.* at 1384. In stark contrast with its decision in *Oracle*, the Federal Circuit in *Hutchins* endorsed the First Circuit’s ruling in *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). *See Hutchins*, 492 F.3d at 1383.

In *Borland*, Lotus claimed that a competitor infringed copyright by copying the SSO of the Lotus 1-2-3 command hierarchy for use in the emulation interface of Borland’s Quattro Pro program. *See*

*Borland*, 49 F.3d at 810. Borland argued that it used the same commands in the same order so that users who had constructed macros of frequently executed functions in Lotus could continue to use those macros in the Borland program. *See id.* Those macro programs, in other words, would not interoperate with the Borland program unless the emulation mode commands were in exactly the same order.

As in *Altai*, the First Circuit did not find the SSO concept helpful in distinguishing protectable and unprotectable structural elements of computer programs. It held, rightly in our view, that the SSO at issue in *Borland* was an unprotectable method of operation under Section 102(b), akin to the command structure of VCR machines. *See id.* at 815-17. The First Circuit's *Borland* decision has attracted considerable support in the law review literature since this Court's 4-4 split in that case.<sup>4</sup> The caselaw

---

<sup>4</sup> *See, e.g.*, Christina Bohannon, *Reclaiming Copyright*, 23 *Cardozo Arts & Ent. L.J.* 567, 592-93 (2006); Michael W. Carroll, *One For All: The Problem of Uniformity Cost in Intellectual Property Law*, 55 *Am. U. L. Rev.* 845, 899 n.254 (2006); Thomas F. Cotter, *The Procompetitive Interest in Intellectual Property Law*, 48 *Wm. & Mary L. Rev.* 483, 510 n.115 (2006); Stacey L. Dogan & Joseph P. Liu, *Copyright Law and Subject Matter Specificity: The Case of Computer Software*, 61 *N.Y.U. Ann. Surv. Am. L.* 203, 211-12 (2005); Herbert Hovenkamp, *Response: Markets in IP and Antitrust*, 100 *Geo. L.J.* 2133, 2144 n.54 (2012); Dennis S. Karjala, *A Coherent Theory for the Copyright Protection of Computer Software and Recent Judicial Interpretations*, 66 *U. Cin. L. Rev.* 53, 105-107 (1997); Peter Lee, *The Evolution of Intellectual Infrastructure*, 83 *Wash. L. Rev.* 39, 84-85 (2008); Aaron K. Perzanowski, *Rethinking Anticircumvention's Interoperability Policy*, 42 *U.C. Davis L. Rev.* 1549, 1563 n.39 (2009); Randal C. Picker, *Competition and Privacy in Web 2.0 and the Cloud*, 103 *Nw. U. L. Rev. Colloquy* 1, 8 (July 28, 2008),

about the *Borland* decision is split.<sup>5</sup> Accordingly, the *Oracle* case provides an opportunity for this Court to resolve the long-simmering debate over the extent to which copyright protects program SSO.

The lesson of *Altai* and its progeny is that courts must endeavor to distinguish between program structures that are protectable expression and those that are not. Conventional literary works such as Harry Potter novels embody a higher quantum of original expression and fewer unprotectable methods and procedures. Computer programs embody many unprotectable elements, including procedures, processes, systems, methods of operation, abstract ideas and applied know-how. This is why computer programs have a relatively “thin” or “weak” level of copyright protection. *See, e.g., Altai*, 982 F.2d at 712; *Sega*, 977 F.2d at 1527.

### III. When Computer Program Interfaces Constrain the Design Choices of Subsequent Programmers, the Merger

---

<http://www.law.northwestern.edu/lawreview/colloquy/2008/25/>;  
J.H. Reichman & Jonathan A. Franklin, *Privately Legislated Intellectual Property Rights: Reconciling Freedom of Contract with Public Good Uses of Information*, 147 U. Pa. L. Rev. 875, 894 n.70 (1999); Philip J. Weiser, *The Internet, Innovation, and Intellectual Property Policy*, 103 Colum. L. Rev. 534, 604-08 (2003).

<sup>5</sup> *See, e.g., Hutchins v. Zoll Med. Corp.*, 492 F.3d 1377, 1383 (Fed. Cir. 2007), *aff'g* 430 F. Supp. 2d 24 (D. Mass. 2006) (approving district court’s application of *Lotus*); *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366 (10th Cir. 1997) (declining to follow *Lotus*); *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532 (11th Cir. 1996) (noting disagreement between *Lotus* and *Gates* courts on when to consider challenges to originality).



### **Doctrine Precludes Copyright Protection for the Interface Design.**

The Federal Circuit held that as long as Oracle's engineers were not constrained in their design choices as they developed the Java APIs at issue in this case, as long as they exercised creative judgments in selecting and arranging the structure and other components of each API, and as long as these designs satisfy the minimal creativity requirement for copyright protection, then both the Java APIs at issue and their SSO are protectable by copyright law. *See Oracle*, 750 F.3d at 1361.<sup>6</sup> The software copyright caselaw of the last twenty-two years plainly demonstrates that this view is erroneous and requires review by this Court.

After a full trial, the District Court found that the Java APIs at issue in this case were necessary for achieving interoperability. *Oracle*, 872 F. Supp. 2d at 979-81. Accordingly, under *Altai* and *Sega*, the Java APIs should be deemed unprotectable by copyright law under controlling Ninth Circuit precedent.

*Altai* established, and other courts later followed, the rule that external factors such as the "compatibility requirements of other programs with which a program is designed to operate" limit the scope of copyright in programs because these factors constrain the freedom of design choices of subsequent programmers. *Altai*, 982 F.2d at 709-10.

---

<sup>6</sup> Yet the Federal Circuit admitted "[i]t seems possible that the merger doctrine . . . would exclude the three packages identified by the district court as core packages . . . if [programmers] . . . wanted to write in the Java language." *Id.* at 1362.

To interoperate with existing programs, any new program must send and be designed to receive information in the precise fashion required by the interface specifications of the programs with which it is to be compatible. Anyone who develops an API is, in a very real sense, designing that aspect of the program for itself and for others.

The Second Circuit in *Altai* was convinced that Altai had taken from CA's program only what was necessary to achieve compatibility. *See id.* at 714-15. The outcome in the Federal Circuit's decision in *Atari Games* was different because the defendant copied more than was necessary to achieving compatibility with Nintendo's programs. *See Atari Games*, 975 F.2d at 843.

The Second Circuit in *Altai* referred to the merger doctrine in discussing why external factors such as compatibility needs limit the scope of copyright protection in programs. *See Altai*, 982 F.2d at 708-09. That is a sound doctrinal basis for such a ruling, and the District Court in this case properly applied it. Courts often recognize that when there is only one or a small number of ways to express an idea, idea and expression will be considered to have merged, and no copyright protection is available to the merged elements. *See, e.g., Lexmark*, 387 F.3d 522, 535-36 (6th Cir. 2004) (“[In considering whether merger and scenes a faire doctrines apply,] [t]he question . . . is not whether *any* alternatives theoretically exist; it is whether other options practically exist under the circumstances.”). *But see, e.g., M. Kramer Mfg. Co. v. Andrews*, 783 F.2d 421, 436 (4th Cir. 1986) (finding plaintiff's video game not subject to merger if other programs could be written

to perform the same function) (*citing Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983).

The merger principle, like Section 102(b)'s exclusion of methods and processes, derives from *Baker v. Selden*. This Court observed that “where the [useful] art [a work] 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.” *Baker*, 101 U.S. at 103. Baker had no choice but to use substantially the same arrangement of headings and columns if he wanted to reimplement Selden’s bookkeeping system in his own independently written work. The Java APIs at issue in this case pose similar constraints on the design choices of subsequent programmers.

Ninth Circuit rulings that should have been given considerable deference on this appeal reached the same conclusion about the unprotectability of interfaces necessary for interoperability. It did not matter in *Sega* that the plaintiff had designed the interface procedures for its Genesis console in a creative way. The Sega interface procedures constrained Accolade’s design choices when it sought to write a program that would run on the Sega platform. *See Sega*, 977 F.2d at 1522. For this reason, the interface procedures were unprotected aspects of the Sega program under Section 102(b). *See id.* at 1526. Nor did it undercut Accolade’s defense that Sega had a licensing program for Genesis-compatible videogames in which Accolade declined to participate. *See id.* at 1514, 1523.

Eight years after *Sega*, the Ninth Circuit revisited the legality of reverse-engineering copyrighted program code, considering this time the development of software that could interoperate with third-party software designed to run on the plaintiff's platform. In *Sony Computer Entertainment, Inc. v. Connectix, Corp.*, 203 F.3d 596 (9th Cir. 2000), Connectix developed a program to emulate the functionality of the Sony PlayStation. Sony sought to distinguish *Sega* on numerous grounds, all of which were unavailing. *See id.* at 602-07. Sony interface procedures, no matter how creative they might have been at the outset, were unprotected elements of the PlayStation software, *see id.* at 603, and reverse engineering Sony's code to get access to these unprotected elements was fair use. *See id.* at 608. It did not matter that the Connectix software aimed to be a substitute for the plaintiff's product, *see id.* at 606-07, and not merely a complementary product as in *Sega*. *See Sega*, 977 F.2d at 1522. Nor, apparently, did it matter that the Connectix software was not fully compatible with the PlayStation games. *See Connectix*, 203 F.3d at 599.

The only decision—other than the Federal Circuit ruling in *Oracle*—to cast doubt on the lack of copyright protection for computer program elements required for interface compatibility was the Third Circuit's in *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983). Apple claimed copyright infringement because Franklin copied the Apple operating system (OS) programs in making a computer compatible with programs written to run on the Apple II computer. *See Apple*, 714 F.2d at 1243-44. Franklin asserted that it was

necessary to copy the Apple OS in order to be compatible with applications software developed to run on the Apple platform, for there were only “a limited number of ways to arrange operating systems to enable a computer to run . . . Apple-compatible software.” *Id.* at 1253 (citation omitted) (internal quotation marks omitted).

The Third Circuit regarded Franklin’s compatibility argument as having “no pertinence to either the idea/expression dichotomy or merger.” *Id.* Compatibility was, in its view, “a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged.” *Id.*

This dicta should be given no weight for three reasons. First, Franklin made no effort to reimplement the interface procedures embedded in the Apple OS in independently written code. It made exact copies of the Apple programs. *See id.* at 1245. Second, these statements were made at an early stage in the evolution of software copyright law, well before the *Altai*, *Atari Games*, *Sega*, and *Connectix* cases provided more thorough analyses of the copyright implications of a second comer’s reimplementation of interface procedures necessary for interoperability. Third, the very purpose of developing and promoting widespread use of Java APIs was to enable greater interoperability among programs. Under the rules of Java, as the District Court recognized, a programmer must use identical declaration or method header lines “to declare a method specifying the *same* functionality.” *Oracle*, 872 F. Supp. 2d at 976.

The District Court gave appropriate weight to the later circuit court rulings and wisely eschewed the anti-compatibility dicta from the earlier *Franklin* decision.

If this Court does not repudiate the Federal Circuit's erroneous interpretation of the merger doctrine as applied to computer programs, the result will likely be a surge in litigation over the protectability of API code required for program interoperability, even though this issue had seemed to be resolved by appellate rulings going back to 1992. Based on these rulings, innovators have relied on the understanding that computer code that establishes compatibility between programs is unprotectable, and competition and innovation in the U.S. software industry has thrived in part because of innovators' ability to rely on these rulings in creating APIs.

The decision below has opened up a new period of uncertainty about the copyrightability of APIs. Unless reversed, this is likely to have a chilling effect on the willingness of software startup firms and open source developers to create innovative programs that can be executed on incumbent firm platforms. Competition and innovation in the software industry will suffer unless this Court resolves the circuit split on this issue and repudiates the Federal Circuit's ruling.

## CONCLUSION

The Federal Circuit decision in the *Oracle* case is at odds with Ninth Circuit precedent and with more than two decades of copyright jurisprudence

concerning the application of copyright law to computer programs. It erred in interpreting Section 102(b) of the 1976 Act, in holding that the Java APIs at issue in the case are protectable SSO, and in misconstruing the merger doctrine. In order to remedy this split and reaffirm traditional principles of copyright law as applied to computer programs, the petition for writ of certiorari should be granted.

Respectfully submitted,

PAMELA SAMUELSON  
*Counsel of Record*  
University of California,  
Berkeley, School of Law  
434 Boalt Hall  
Berkeley, California 94720  
(510) 642-6775  
pam@law.berkeley.edu  
*Counsel of Record for Amici*  
*Curiae Intellectual*  
*Property Professors*

November 7, 2014

**APPENDIX: List of Signatories**

(Institutions are listed for identification purposes only)

Timothy K. Armstrong, Professor  
University of Cincinnati College of Law

Clark D. Asay, Associate Professor  
Brigham Young University Law School

Oren Bracha, Professor  
University of Texas School of Law

Annemarie Bridy, Professor  
University of Idaho College of Law

Dan Burk, Professor  
University of California, Irvine, School of Law

Michael A. Carrier, Professor  
Rutgers Law School – Camden

Brian W. Carver, Assistant Professor  
University of California, Berkeley  
School of Information

Bernard Chao, Assistant Professor  
University of Denver Sturm College of Law

Margaret Chon, Professor  
Seattle University School of Law

Ralph D. Clifford, Professor  
University of Massachusetts School of Law



Julie E. Cohen, Professor  
Georgetown University Law Center

Ben Depoorter, Professor  
University of California, Hastings College of the Law

Roger Allan Ford, Assistant Professor  
University of New Hampshire School of Law

Shubha Ghosh, Professor  
University of Wisconsin Law School

Eric Goldman, Professor  
Santa Clara University School of Law

James Grimmelman, Professor  
University of Maryland Francis King Carey School of  
Law

Paul Heald, Professor  
University of Illinois College of Law

Peter Jaszi, Professor  
American University, Washington College of Law

Dennis S. Karjala, Professor  
Sandra Day O'Connor College of Law, Arizona State  
University

Marshall Leaffer, Distinguished Scholar in IP Law  
and University Fellow  
Indiana University Maurer School of Law

Mark A. Lemley, Professor  
Stanford Law School

Yvette Joy Liebesman, Associate Professor  
Saint Louis University School of Law

Brian J. Love, Assistant Professor  
Santa Clara University School of Law

Glynn Lunney, Jr., Professor  
Tulane University School of Law

Michael J. Madison, Professor  
University of Pittsburgh School of Law

Stephen McJohn, Professor  
Suffolk University Law School

Michael J. Meurer, Professor  
Boston University School of Law

Lateef Mtima, Professor  
Howard University School of Law

Ira Steven Nathenson, Professor  
St. Thomas University School of Law

Tyler T. Ochoa, Professor  
Santa Clara University School of Law

Mark R. Patterson, Professor  
Fordham University School of Law

Aaron Perzanowski, Associate Professor  
Case Western Reserve University School of Law

Jerome Reichman, Professor  
Duke University School of Law

Michael Risch, Associate Professor  
Villanova University School of Law

Michael L. Rustad, Professor  
Suffolk University Law School

Matthew Sag, Professor  
Loyola University Chicago School of Law

Zahr Said, Assistant Professor  
University of Washington School of Law

Jessica Silbey, Professor  
Suffolk University Law School

Christopher Jon Sprigman, Professor  
New York University School of Law

Rebecca Tushnet, Professor  
Georgetown University Law Center

Jennifer M. Urban, Assistant Clinical Professor  
University of California, Berkeley, School of Law