API COPYRIGHTABILITY BLEAK HOUSE: UNRAVELING AND REPAIRING THE *ORACLE V. GOOGLE* JURISDICTIONAL MESS

Peter S. Menell†

ABSTRACT

Like Dickens' tale of Jarndyce and Jarndyce, the *Oracle v. Google* litigation has droned on for what seems like generations in the software industry with no clear end in sight. The litigation is on an especially wasteful and perilous course due to its peculiar jurisdictional posture. As a result of patent infringement allegations lodged in the complaint, the Federal Circuit has exclusive appellate jurisdiction notwithstanding that neither party appealed the rejection of the patent causes of action. Hence, the only issues presented to the Federal Circuit were copyright issues governed by Ninth Circuit—as opposed to Federal Circuit—jurisprudence. The Federal Circuit misinterpreted Ninth Circuit (and general) copyright law, thereby steering the case into a needless fair use retrial.

Congress did not provide a mechanism short of Supreme Court review for ensuring that the Federal Circuit properly interpreted regional circuit law. After tracing the history of the *Oracle v. Google* litigation and critiquing the Federal Circuit's analysis, this Article evaluates a range of potential reforms to the appellate jurisdictional mess presented by software intellectual property litigation and proposes several solutions to this Dickensian predicament.

DOI: https://dx.doi.org/10.15779/Z38B56D426

† Koret Professor of Law and Director of the Berkeley Center for Law & Technology, University of California at Berkeley School of Law. I am grateful to Mark Lemley, Tejas Narechania, David Nimmer, Paul Gugliuzza, and Ryan Vacca for comments on this project, and to Andrea Hall for research assistance.

TABLE OF CONTENTS

I. **INTRODUCTION** 1517

II. **THE *ORACLE V. GOOGLE* LITIGATION: FROM MICROCOMPUTERS TO THE INTERNET AGE** 1520

A. LEGISLATIVE AND JURISPRUDENTIAL BACKDROP 1520

1. *Copyright Legislation* 1520
2. *Copyright Jurisprudence* 1523

B. ROOTS OF THE *ORACLE V. GOOGLE* LITIGATION 1532

1. *The Java Platform* 1533
2. *The Android Platform* 1538

C. THE *ORACLE V. GOOGLE* LITIGATION 1549

1. *The 2012 Trial* 1550
2. *Federal Circuit Reversal* 1555
3. *The Interlocutory Certiorari Petition* 1558
4. *The 2016 Fair Use Retrial* 1559

III. **CRITIQUE OF THE FEDERAL CIRCUIT'S 2014 COPYRIGHTABILITY DECISION** 1562

A. MISINTERPRETATION OF THE COPYRIGHT ACT 1562

B. MISREADING NINTH CIRCUIT JURISPRUDENCE 1564

C. CONFLATION OF EXPRESSIVE AND TECHNOLOGICAL "CREATIVITY" 1567

D. OVERLY RIGID APPROACH TO LIMITING DOCTRINES 1567

E. TREATING API DESIGN AS VARIABLE EXPRESSION RATHER THAN UNIQUE FUNCTION 1570

IV. **THE *ORACLE V. GOOGLE* LITIGATION: FUTURE PATHWAYS** 1571

V. **DEBUGGING APPELLATE INTELLECTUAL PROPERTY JURISDICTION** 1576

A. THE FEDERAL CIRCUIT'S SUBJECT MATTER JURISDICTION 1578

B. ANALYTICAL FRAMEWORK FOR ASSESSING APPELLATE INTELLECTUAL PROPERTY JURISDICTION 1581

1. *Jurisprudential Integrity* 1582
2. *Federalism* 1583
3. *Specialization Bias* 1587
4. *Administrative Efficiency* 1590

C. REFINING APPELLATE INTELLECTUAL PROPERTY JURISDICTION 1591

1. *District Court Case Management and Routing of Appellate Review* 1591
2. *Appellate Jurisdiction Reforms* 1593

VI. **CONCLUSIONS** 1595

I. INTRODUCTION

For more than six years, Oracle and Google have fought a costly and—as of this writing—still unresolved battle over copyright protection for application program interfaces (APIs).[1] The dispute has significant ramifications for much of the software industry,[2] which has been drawn into a high technology version of Jarndyce and Jarndyce. As Charles Dickens explained in the opening chapter of *Bleak House*:

> Jarndyce and Jarndyce drones on. This scarecrow of a suit has, in course of time, become so complicated, that no man alive knows what it means. The parties to it understand it least; but it has been observed that no two Chancery lawyers can talk about it for five minutes, without coming to a total disagreement as to all the premises . . . but Jarndyce and Jarndyce still drags its dreary length before the Court, perennially hopeless.[3]

[1] An API is:

> a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. It defines methods of communication between various software components and provides access to data of an operating system, application, or other service. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware or software library. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables or remote calls.

Application Programming Interface, WIKIPEDIA, https://en.wikipedia.org/wiki/Application_programming_interface (last visited Oct. 8, 2017); Lothar Determann & David Nimmer, *Software Copyright's* Oracle *From the Cloud*, 30 BERKELEY TECH. L.J. 161, 170 (2015) (describing litigation between Google and Oracle over APIs).

[2] *See* Nick Wingfield & Quentin Hardy, *Google Prevails as Jury Rebuffs Oracle in Code Copyright Case*, N.Y. TIMES (May 26, 2016), http://www.nytimes.com/2016/05/27/technology/google-oracle-copyright-code.html (quoting representatives of the Electronic Frontier Foundation, Public Knowledge, and a venture capital firm praising the jury's verdict); 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/ ("Software developers routinely treat APIs as exempt from copyright protection"); Don Clark & Cari Tuna, *Oracle Suit Challenges Google–Silicon Valley Giants Tangle Over Patents, Copyrights Involving Open Programs Android and Java*, WALL ST. J., Aug. 13, 2010, at B1 (noting that the lawsuit was a "surprise move" and "set off shock waves in the Silicon Valley software community").

[3] CHARLES DICKENS, BLEAK HOUSE 3 (1853).

Although the *Oracle v. Google* litigation has not yet gone on as long as Jarndyce and Jarndyce in human years, it spans several software generations. Software years are more like dog years.[4] The industry evolves so quickly that companies wither and die if they do not continually innovate. Building on and interoperating with widely adopted software platforms is the lifeblood of Internet age computing and commerce. Yet the *Oracle v. Google* litigation looms, like a dark cloud, over the industry.

The *Oracle v. Google* litigation is on a wasteful and perilous course due to its peculiar jurisdictional posture. In what seems especially ironic in the context of this litigation, Congress established the U.S. Court of Appeals for the Federal Circuit in 1982 for the express purpose of "ending the current legal confusion created by eleven different appellate forums, all generating different interpretations of the patent law."[5] Congress addressed the problem by granting the Federal Circuit exclusive jurisdiction over patent appeals. Yet the Federal Circuit's exclusive appellate jurisdiction over cases involving patent infringement allegations has created a new species of interpretive confusion. In patent cases that contain copyright (or other non–patent) causes of action, regional circuit law binds the Federal Circuit's review of legal questions not exclusively assigned to the Federal Circuit.[6] Moreover, the Federal Circuit will hear the appeals of such non–patent issues even if, as was the circumstance in *Oracle v. Google*, neither party challenged the district court's patent rulings.[7] Congress did not provide a mechanism short of Supreme Court review for ensuring that the Federal Circuit properly interpreted regional circuit law.

[4] In popular lore, one dog year is the equivalent to seven human years. *See* Erika Mansourian, *How to Calculate Dog Years to Human Years*, AM. KENNEL CLUB (Nov. 16, 2015), http://www.akc.org/content/entertainment/articles/how-to-calculate-dog-years-to-human-years/. The American Veterinary Medical Association offers a more sophisticated formula. The first dog year is equivalent to fifteen human years. The second dog year is equivalent to nine human years. Each additional dog year is equivalent to five human years. *Senior Pets*, AM. VETERINARY MED. ASS'N, https://www.avma.org/public/PetCare/Pages/Senior-Pets.aspx (last visited Oct. 8, 2017).

[5] *See* H.R. REP. NO. 96-1307 (1980) (commenting on the legislation that would become the Federal Courts Improvement Act of 1981, Pub. L. 97–164, 96 Stat. 25); *see also* COMM'N ON REVISION OF THE FED. COURT APPELLATE SYS., STRUCTURE AND INTERNAL PROCEDURES: RECOMMENDATIONS FOR CHANGE 15, *as reprinted in* 67 F.R.D. 195, 220 (1975) (discussing the problem of forum shopping in the context of patent cases) [hereinafter HRUSKA COMMISSION REPORT].

[6] *See id*. Copyright issues are not exclusively assigned to the Federal Circuit. *See* 28 U.S.C. § 1295 (2012).

[7] *See* Atari Games Corp. v. Nintendo of Am., Inc., 897 F.2d 1572, 1575 (Fed. Cir. 1990).

For reasons summarized in this Article and explored in greater depth in a parallel project,[8] the Federal Circuit's 2014 decision in *Oracle v. Google* misinterpreted Ninth Circuit law (and copyright law in general). The unusual jurisdictional posture of the *Oracle v. Google* litigation has produced a Gordian knot of Federal Circuit/Ninth Circuit copyright jurisprudence that cannot easily be untied. Due to the path dependence of the litigation, it is unclear whether the core API copyrightability issue will ever be ripe for Supreme Court review.

Even as *Oracle v. Google* heads back to the Federal Circuit for needless review of a needless second trial, another major software copyright battle governed by this mutant jurisprudence is unfolding in another Northern District of California courtroom.[9] In 2014, Cisco Systems, a leading manufacturer of networking equipment, sued Arista Networks for patent and copyright infringement.[10] As in the *Oracle v. Google* litigation, the copyright claims focus on alleged infringement of Cisco's command line interface (CLI) for configuring, monitoring, and maintaining Cisco devices—an API copyright claim.[11] The district judge, in that case, faces a dilemma—whether to follow the Ninth Circuit's jurisprudence or the Federal Circuit's interpretation of the Ninth Circuit's jurisprudence. In the *Oracle* case, District Judge William Alsup fell into this trap. The result was a reversal of his copyrightability determination. The software industry at large faces a similar dilemma.

This Article examines how software copyright jurisprudence has arrived at this precarious state as well as the larger ramifications for the software industry and appellate intellectual property jurisdiction. Part II summarizes the long and winding history of the *Oracle v. Google* litigation. Part III critiques the Federal Circuit's 2014 copyrightability decision. Part IV traces the possible future pathways for the litigation and explains why the confusing cloud of copyright jurisprudence might continue to loom over the

[8] *See* 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. (forthcoming 2018).

[9] *See* Quentin Hardy, *In Suit, Cisco Accuses Arista of Copying Work*, N.Y. TIMES: BITS (Dec. 5, 2014), http://bits.blogs.nytimes.com/2014/12/05/in-suit-cisco-accuses-arista-of-copying-work/.

[10] *See id.*

[11] *See* Cisco Sys. v. Arista Networks, Inc., No. 14-cv-05344-BLF, 2016 U.S. Dist. LEXIS 113285 (N.D. Cal. Aug. 23, 2016); Scott Graham, Cisco v. Arista *IP Battle Starts to Look a Lot Like* Oracle v. Google, RECORDER (Sept. 14, 2017), http://www.therecorder.com/id=1202766017854/Cisco-v-Arista-IP-Battle-Starts-to-Look-a-Lot-Like-Oracle-v-Google.

software industry. Part V explores ways of repairing appellate jurisdiction. Part VI concludes.

II. THE *ORACLE V. GOOGLE* LITIGATION: FROM MICROCOMPUTERS TO THE INTERNET AGE

The *Oracle v. Google* litigation emerged from a dynamic industrial saga that in many ways reflects the evolution of the modern software industry. Section A presents the legislative and jurisprudential backdrop. Section B explores the development of the Java programming language and platform as well as the Android mobile platform. Section C traces the first six years of the *Oracle v. Google* litigation.

A. LEGISLATIVE AND JURISPRUDENTIAL BACKDROP

This Section summarizes the legislative and jurisprudential background to the *Oracle v. Google* litigation.

1. Copyright Legislation

The *Oracle v. Google* saga traces back to Congress's equivocal decision to bring computer software within the scope of copyright protection. Computer software could be expensive to develop and was easily pirated, creating a severe appropriability problem for the nascent, yet critical, commercial software industry.[12] Patent law has long served as the primary intellectual property regime for technological advance.[13] By contrast, copyright law serves as the principal mode of protection for aesthetic creativity.[14] Although computer software—functioning as the gears and levers for digital machines—fell within the technological as opposed to the aesthetic arts, its textual form could more easily be protected through a copyright–type regime, which had long been the primary means of limiting piracy of literary works. Copyright's low threshold for protection, complex scope of protection, broad array of rights, and long duration, however, risked overprotecting software and thereby undermining technological innovation and competition.

[12] *See generally* Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329 (1987).

[13] *See* PETER S. MENELL, MARK A. LEMLEY, & ROBERT P. MERGES, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE: 2017, VOL I: PERSPECTIVES, TRADE SECRETS, AND PATENTS 168 (2017).

[14] *See* PETER S. MENELL, MARK A. LEMLEY, & ROBERT P. MERGES, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE: 2017, VOL II: COPYRIGHTS, TRADEMARKS & STATE IP PROTECTIONS 498–500 (2017).

The software protection controversy emerged at an inopportune time. Congress had been working for nearly two decades to overhaul the Copyright Act of 1909 and was nearing closure in the early to mid–1970s.[15] Faced with the difficult challenge of fitting computer and other new information technologies under the existing umbrella of intellectual property protection, Congress established the National Commission on New Technological Uses of Copyrighted Works (CONTU) to study the implications of computer software and recommend revisions to federal intellectual property law.[16]

As a stopgap, Congress included software within the scope of copyright protection in the Copyright Act of 1976 ("1976 Act"),[17] but subject to foundational limitations set forth in § 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."[18] The legislative history noted that:

> [s]ome 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.[19]

After conducting extensive hearings and receiving expert reports, a majority of CONTU's blue–ribbon panel of copyright authorities and interest group representatives concluded that the intellectual work

[15] *See* Peter S. Menell, *In Search of Copyright's Lost Ark: Interpreting the Right to Distribute in the Internet Age*, 59 J. COPYRIGHT SOC'Y U.S.A. 1, 31–32 (2011).

[16] Act of Dec. 31, 1974, Pub. L. No. 93-573, § 201, 88 Stat. 1873.

[17] The Act includes "literary works" within the class of "works of authorship." *See* 17 U.S.C. § 102(a)(1) (2012). 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).

[18] 17 U.S.C. § 102(b) (2012).

[19] *See* H.R. REP. NO. 94-1476, at 57 (1976).

embodied in computer software should be protected under copyright law, notwithstanding the fundamental principle that copyright cannot protect "any idea, procedure, process, system, method of operation, concept, principle, or discovery"[20] and the Supreme Court's foundational decision in *Baker v. Selden*.[21] CONTU recommended two modest changes to the 1976 Act: (1) defining a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result"; and (2) allowing "the rightful possessor of a copy of a computer program" to run the program and to make a backup copy of the program without infringement liability.[22] Congress implemented CONTU's recommendation in its 1980 amendments to federal copyright law with one confusing wording change.[23]

The CONTU Final Report explained 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—originality and the idea–expression dichotomy.[24] The Report 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

[20] 17 U.S.C. § 102(b) (2012).

[21] *See* NAT'L COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 1 (1979), http://digital-law-online.info/CONTU/PDF/index.html [hereinafter CONTU FINAL REPORT]. *But see id.* at 27–37 (Commissioner Hersey, dissenting) (arguing that "forcible wrenching" would be required to protect computer programs under the copyright law); *id.* at 37–38 (Commissioner Karpatkin, dissenting) (same); *cf. id.* at 26–27 (Commissioner Melville B. Nimmer, concurring) (warning that CONTU recommendations might take copyright law "beyond the breaking point," converting it into a general misappropriation law).

[22] *See id.* at 12.

[23] *See* Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3007, 3028 (codified at 17 U.S.C. §§ 101, 117 (2012)). For reasons that were not explained in the legislative history of the 1980 amendments, Congress narrowed CONTU's category of "rightful possessor" to "rightful owner." *See* 2 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.08[B][1][c][ii] (2017).

[24] *See* CONTU FINAL REPORT, *supra* note 21, at 20. Courts have treated the CONTU FINAL REPORT as legislative history for the 1980 amendments to the 1976 Act. *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).

> given task, their later use by another will not amount to an infringement.[25]

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 programming design and coding choices to the extent that the expression was separable from the underlying ideas. In this way, the general programming ideas and unoriginal programming choices remain free for others to use while the creative effort in particularized programming choices and compilations, especially in complex programs, gains protection from copyists.

Interpreting and applying the idea–expression dichotomy in software cases, like other important "common law" copyright doctrines,[26] fell to the courts.

2. *Copyright Jurisprudence*

The rapid growth of the microcomputer and consumer software industries fueled more than a decade of litigation centered on the scope of copyright protection for computer software. These cases spanned Apple's litigation to bar clones of its breakthrough Apple II computer, Apple's effort to block Microsoft Windows from competing with the Macintosh's graphical user interface, mobile phone companies' copyright claims to codes for cellular phone networks, Sega's effort to control access to its Genesis videogame console, and Lotus's effort to control the menu command hierarchy of the Lotus 1–2–3 spreadsheet program. These cases, and many other software copyright battles, centered on the idea–expression dichotomy: to what extent could platform innovators protect application program interfaces through copyright protection?

The early cases suggested a broad scope of copyright protection for computer software and interoperable features. The first major software copyright cases pitted Apple Computer Corporation, then a young, break–out microcomputer company, against brash competitors offering inexpensive "interoperable" Apple II clones.[27] The clone makers quickly entered the market by copying, bit by bit, Apple's operating system and

[25] *Id*. at 20 (footnote omitted).

[26] *See generally* Peter S. Menell, *The Mixed Heritage of Federal Intellectual Property Law and Ramifications for Statutory Interpretation*, *in* INTELLECTUAL PROPERTY AND THE COMMON LAW 63 (Shyamkrishna Balganesh ed., 2013).

[27] *See* Apple Comput., Inc. v. Franklin Comput. Corp., 545 F. Supp. 812 (E.D. Pa. 1982), *rev'd*, 714 F.2d 1240 (3d Cir. 1983); Apple Comput., Inc. v. Formula Int'l, Inc., 562 F. Supp. 775 (C.D. Cal. 1983), *aff'd*, 725 F.2d 521 (9th Cir. 1984).

application programs. The defendants argued that copyright protection did not extend to non–human readable (object code) formats of computer software and that the idea–expression doctrine barred copyright protection for operating system programs. They further argued that copyright protection should not stand in the way of selling computers that can run programs written for the Apple II.

The courts had little difficulty finding that copying the entirety of sophisticated computer programs constituted copyright infringement. In reaching these findings, however, the courts went overboard in their dicta. Addressing the defendant's interoperability argument, the Third Circuit opined 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."[28] Since two entirely different programs can achieve the same "certain result[s]"—for example, generate the same set of protocols needed for interoperability—the court was not justified in making such an expansive statement about the scope of copyright protection for computer program elements.[29] Given the verbatim copying of millions of bits of object code, there was no need to address the interoperability issue. The defendant offered no explanation of which elements of the program were protectable and which were not.

The next major software copyright appellate decision also arose from the Third Circuit. In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*,[30] a computer programmer sued the dental laboratory, for which it had developed a computer program for managing its bookkeeping functions, for copyright infringement after an officer of the laboratory set out to create a version of the program that would run on other computer systems. The competing software did not literally copy Whelan's code, but there were overall structural similarities between the two programs. To distinguish protectable expression from unprotectable idea, the court reasoned:

> *[T]he 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

[28] *See* Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1253 (3d Cir. 1983).

[29] CONTU was clear that "[o]ne is always free to make the machine do the same thing as it would if it had the copyrighted work placed in it, but only by one's own creative effort rather than by piracy." *See* CONTU FINAL REPORT, *supra* note 21, at 21.

[30] 797 F.2d 1222 (3d Cir. 1986).

> 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.[31]

In applying this rule, the court defined the idea as "the efficient management of a dental laboratory," which could be expressed in countless ways.[32] Drawing the idea–expression dichotomy at such a high level of abstraction implies an expansive scope of copyright protection. Furthermore, the court's conflation of merger analysis and the idea–expression dichotomy implicitly allows copyright protection of procedures, processes, systems, and methods of operation that § 102(b) expressly excludes.

Although the case did not directly address copyright protection for interoperability protocols, the court's mode of analysis dramatically expanded the scope of copyright protection for computer programs. If everything below the general purpose of the program was protectable under copyright, then it would follow that particular protocols were protectable because there would be alternative means to accomplish the program's general purpose. Such a result would effectively bar competitors from developing interoperable programs and computer systems.

Commentators roundly criticized the *Whelan* test,[33] and other courts began to refine the scope of copyright protection to comport with the fundamental principles (including limitations) of copyright protection. A few months after the *Whelan* decision, the Fifth Circuit confronted a similar claim of copyright infringement based upon structural similarities between two programs designed to provide cotton growers with information regarding cotton prices and availability, accounting services, and a means for conducting cotton transactions electronically.[34] In declining to follow the *Whelan* approach, the court found that the similarities in the programs

[31] *Id.* at 1236 (emphasis in original) (citations omitted).

[32] *Id.*

[33] *See, e.g.*, Donald S. Chisum et al., *LaST Frontier Conference Report on Copyright Protection of Computer Software*, 30 JURIMETRICS J. 15, 20–21 (1989); Steven R. Englund, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 881 (1990); Peter S. Menell, *Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045 (1989); David Nimmer, Richard L. Bernacchi & Gary N. Frischling, *A Structured Approach to Analyzing the Substantial Similarity of Computer Software in Copyright Infringement Cases*, 20 ARIZ. ST. L.J. 625, 629–34 (1988).

[34] Plains Cotton Coop. Ass'n v. Goodpasture Comput. Serv., Inc., 807 F.2d 1256 (5th Cir. 1987).

were dictated largely by standard practices and forms in the cotton market—what the court called "externalities"—which constitute unprotectable ideas.[35]

In 1992, the Second Circuit adapted Learned Hand's seminal abstraction–filtration–comparison framework[36] to computer software analysis.[37] Computer Associates (CA), a leading mainframe software provider, had developed SCHEDULER, a job–scheduling program that worked with three IBM mainframe computers. Part of the success of this program was that it had a subcomponent, called ADAPTER, which would interoperate with any of the three IBM mainframes (DOS/VSE, MVS, and VM/CMS). As a result, the user did not need to customize its programs for each of the IBM mainframes. ADAPTER ensured that programs written for SCHEDULER would interoperate with any of the three IBM mainframes.

In developing a competing job scheduling computer program (ZEKE), which had its own code layer (OSCAR) for interoperating with the three IBM mainframes, Altai relied on James Arney, a former CA programmer. Unbeknownst to Altai's management, Arney improperly copied 30% of OSCAR from CA's ADAPTER program.[38] When Altai's executives learned of the illicit copying, the company initiated a clean–room[39] rewrite of the program. Drawing on the *Whelan* decision, CA challenged the revised version of ZEKE based on structural similarities. The district court criticized *Whelan*'s "simplistic test" for determining similarity between computer programs,[40] rejecting the notion that there is but one idea per program and that as long as there were alternative ways of expressing that one idea, then any particular version was protectable under copyright law.

[35] *Id.* at 1262 (finding the commonly used "cotton recap sheet," for summarizing basic transaction information, to be unprotectable). The court was persuaded by the decision in *Synercom Technology, Inc. v. University Computing Co.*, 462 F. Supp. 1003, 1012–13 (N.D. Tex. 1978), which analogized the "input formats" of a computer program (the organization and configuration of information to be inputted into a computer) to the "figure-H" pattern of an automobile stick shift.

[36] *See* Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930).

[37] *See* Comput. Assocs. Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).

[38] *Id.* at 699. Altai accepted responsibility for copyright infringement based on Arney's misdeeds and was ordered to pay $364,444 in damages. *See id.* at 696.

[39] A clean room process insulates programmers from copyright protected code in producing code that accomplishes the same functions as a target program based solely on the functional specifications. Such a process ensures that a program is independently written and hence not copied except with regard to unprotectable elements. *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).

[40] 775 F. Supp. 544, 558 (E.D.N.Y. 1991).

Focusing on the various levels of the computer programs at issue, the court determined that the similarities between the programs were dictated by external factors—such as the interface specifications of the IBM operating system and the demands of functionality—and hence no protected code was infringed.

The Second Circuit decision fleshed out the analytical framework for determining copyright infringement of computer program code:

> In ascertaining substantial similarity . . . a court would first break down the allegedly infringed program into its constituent structural parts. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable material. Left with a kernel, or perhaps kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program.[41]

The court's abstraction–filtration–comparison test recognized that an idea could exist at multiple levels of a computer program and not solely at the most abstract level. Furthermore, the ultimate comparison is not between the programs as a whole but rather between a program's protectable elements and those that allegedly copy them. Of most importance with regard to fostering interoperability, the court held copyright protection did not extend to those program elements where the programmer's freedom to choose is:

> circumscribed by extrinsic considerations such as (1) mechanical specifications of the computer on that a particular program is intended to run; (2) compatibility requirements of other programs with which a program is designed to operate in conjunction; (3) computer manufacturers' design standards; (4) demands of the industry being serviced; and (5) widely accepted programming practices within the computer industry.[42]

[41] *Altai*, 982 F.2d at 706.

[42] *Id*. at 709–10. The court observed that:

> [w]hile, hypothetically, there might be a myriad of ways in which a programmer may effectuate certain functions within a program–*i.e.*, express the idea embodied in a given subroutine–efficiency concerns may so narrow the practical range of choice as to make only one or two forms of expression workable operations.

Id. at 708.

Directly rejecting the dictum in *Apple v. Franklin*,[43] the Second Circuit recognized that external factors such as interface specifications, de facto industry standards, and accepted programming practices are not protectable under copyright law. The Second Circuit test evaluates these external factors at the time of the allegedly infringing activities (i.e., ex post), not at the time that the first program is written.[44]

Commentators warmly embraced the *Altai* decision,[45] and courts have universally adopted the abstraction–filtration–comparison.[46] The Ninth Circuit's decision in *Sega Enterprises Ltd. v. Accolade*[47] expressly recognized the legitimacy of deciphering and copying lockout codes for purposes of developing interoperable products. Sega developed a successful video game platform (Genesis) for which it licensed access to video game developers. Accolade, a manufacturer of video games, wanted to distribute versions of its games on the Genesis platform. It did not, however, want to limit distribution exclusively to Genesis, as Sega required. Rather than license Sega's code, Accolade reverse engineered the access code through a painstaking effort that entailed making hundreds of intermediate copies of Sega's computer code. Accolade then incorporated only the code (approximately 25 bytes in games containing between 500,000 and 1.5 million bytes) necessary to achieve interoperability with the Genesis platform.[48]

Sega sued Accolade for copyright infringement. Given the relatively small amount of Sega code in the Accolade game cartridges, Sega focused its copyright claim on the making of intermediate copies of its full computer program made during the process of reverse engineering. The district court

[43] *See* Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1251 (3d Cir. 1983).

[44] The court emphasized that the first to write a program for a particular application should not be able to "'lock up' basic programming techniques as implemented in programs to perform particular tasks." *Altai*, 982 F.2d at 712 (quoting Menell, *supra* note 33).

[45] *See* David Bender, Computer Associates v. Altai*: Rationality Prevails*, 9 COMPUTER LAW. 1 (1992); Peter S. Menell, *The Challenges of Reforming Intellectual Property Protection for Computer Software*, 94 COLUM. L. REV. 2644, 2652 (1994).

[46] *See* Peter S. Menell, *Envisioning Copyright Law's Digital Future*, 46 N.Y.L. SCH. L. REV. 63, 84–85 (2002); Mark A. Lemley, *Convergence in the Law of Software Copyright?*, 10 BERKELEY TECH. L.J. 1 (1995).

[47] 977 F.2d 1510 (9th Cir. 1992) [hereinafter *Sega Enters. II*].

[48] *Id.* at 1516.

rejected Accolade's argument that such intermediate copies constituted fair use and granted a preliminary injunction.[49]

The Ninth Circuit reversed, holding that "the functional requirements for compatibility with the Genesis [video game console are] aspects of Sega's programs that are not protected by copyright."[50] Building on that foundation, the court ruled that "disassembly of object code in order to gain an understanding of the ideas and functional concepts embodied in the code is a fair use that is privileged by section 107 of the Act."[51] The court determined that policies underlying the Copyright Act authorize disassembly of copyrighted object code and the making of intermediate copies to discover unprotectable elements of code.[52] The Ninth Circuit reaffirmed and expanded this doctrine in *Sony Computer Entertainment, Inc. v. Connectix Corp*.[53]

The Northern District of California and the Ninth Circuit applied the *Altai* framework to the graphical user interface features of a computer program in *Apple Computer, Inc. v. Microsoft Corp*.[54] Apple Computer alleged that Microsoft's Windows operating system infringed Apple's copyrights in the desktop graphical user interface of its Macintosh computer system. A licensing agreement authorizing the defendants' use of aspects of Apple's graphical user interface muddied the copyright issue.[55] The court determined, however, that the licensing agreement was not a complete defense to the copyright infringement claims and consequently analyzed the scope of copyright protection for a range of audiovisual display elements.[56]

The district court found that the unlicensed similarities between Apple's works and Microsoft's Windows were either unprotectable or subject to at least one of copyright law's limiting doctrines. In evaluating the compilation of these elements as a whole, the court applied the "virtual

[49] *See* Sega Enters. Ltd. v. Accolade, Inc., 785 F. Supp. 1392, 1397–1400 (N.D. Cal. 1992), *rev'd*, 977 F.2d 1510 (9th Cir. 1992).

[50] *Sega Enters. II*, 977 F.2d at 1522 (citing 17 U.S.C. § 102(b) (2012)).

[51] *Id.* at 1517–18.

[52] *See id.*

[53] 203 F.3d 596 (9th Cir. 2000).

[54] 799 F. Supp. 1006 (N.D. Cal. 1992), *aff'd in part, rev'd in part*, 35 F.3d 1435 (9th Cir. 1994).

[55] *See id.* at 1015, 1031–32, 1041.

[56] *See* Apple Comput., Inc. v. Microsoft Corp., 759 F. Supp. 1444 (N.D. Cal. 1991); Apple Comput., Inc. v. Microsoft Corp., 717 F. Supp. 1428 (N.D. Cal. 1989); Apple Comput., Inc. v. Microsoft Corp., 709 F. Supp. 925, 930 (N.D. Cal. 1989).

identity" standard[57] and determined that no infringement had occurred. On appeal, the Ninth Circuit affirmed the district court's dissection of Apple's graphical user interface to determine which elements are protectable, filtering of unprotectable elements, and application of the "virtual identity" standard.[58]

The copyrightability of command systems for computer software arose in litigation over spreadsheet technology. Building upon the success of the VisiCalc program developed for the Apple II computer, Lotus Corporation marketed a spreadsheet program for the IBM PC platform—Lotus 1–2–3—which offered many of VisiCalc's features and commands while integrating charting and database capabilities.[59] Lotus 1–2–3 quickly became the market leader for spreadsheets running on IBM and IBM–compatible machines.[60] As a result, knowledge of the program became especially valuable for accountants and managers. The 1–2–3 command hierarchy provided a logically structured menu of more than 200 commands and enabled users to develop customized programs (called macros) for their particular accounting and business planning functions. These investments locked users into the 1–2–3 command structure as their library of macros grew.[61] By the late 1980s, software developers seeking to enter the spreadsheet market could not ignore the large premiums that many consumers placed on transferring their investments in the 1–2–3 system to a new spreadsheet environment, even where a new spreadsheet product offered significant technical improvements over the Lotus spreadsheet.[62]

After three years of intensive development efforts, Borland International, developer of several successful software products including

[57] The Ninth Circuit developed the heightened "virtual identity" standard for evaluating thinly protected works such as compilations of simple, narrowly protected elements, including the visual layout of a day planner (comprising a calendar and ruled lines), *see* Harper House, Inc. v. Thomas Nelson, Inc., 889 F.2d 197 (9th Cir. 1989), and the audiovisual elements of a karate videogame, *see* Data E. USA, Inc. v. Epyx, Inc., 862 F.2d 204 (9th Cir. 1988).

[58] Apple Comput, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994).

[59] *See* D. J. Power, *A Brief History of Spreadsheets*, DSSRESOURCES.COM (Aug. 30, 2004), http://dssresources.com/history/sshistory.html; Pamela Samuelson, *Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of* Lotus v. Paperback, 6 BERKELEY TECH. L.J. 209, 245 n.145 (1991).

[60] *See id.*

[61] *See* Neil Gandal, *Hedonic Price Indexes for Spreadsheets and an Empirical Test for Network Externalities*, 25 RAND J. ECON. 160 (1994).

[62] *See* Mike Hogan, *Product Outlook: Fresh from the Spreadsheet Oven*, PCWORLD, Feb. 1988, at 100, 102; Lawrence J. Magid, *'Surpass' Spreadsheet Program Lives Up to Name, Beats Lotus 1-2-3*, WASH. POST, Apr. 25, 1988, at 26.

Turbo Pascal and Sidekick, introduced Quattro Pro, its entry into the spreadsheet market. Quattro Pro made substantial design and operational improvements and earned accolades in the computer product review magazines.[63] Quattro Pro offered a new interface for its users, which many purchasers of spreadsheets preferred over the 1–2–3 interface. Nonetheless, because of the large number of users already familiar with the 1–2–3 command structure and those who had made substantial investments in developing macros to run on the 1–2–3 platform, Borland considered it essential to offer an operational mode based on the 1–2–3 command structure as well as macro compatibility. Borland's visual representation of the 1–2–3 command mode substantially differed from the 1–2–3 screen displays.

The lower court held that a menu command structure was protectable if there were many such structures available.[64] The court also found that Borland was not permitted to achieve macro compatibility with the 1–2–3 product, distinguishing the treatment of external constraints noted in the *Altai* decision on the ground that such constraints had to exist at the time that the first program was created.[65] The First Circuit reversed, holding that the menu command hierarchy was a "method of operation" that fell within the copyright exclusion set forth in § 102(b).[66] The U.S. Supreme Court

[63] *See Spreadsheet; Borland International Inc.'s Quattro Pro for Windows and Quattro Pro 4.0 for DOS*, PC-COMPUTING, Dec. 1992, at 140 ("No doubt about it: Quattro Pro for DOS is the best DOS spreadsheet there is. Period."); *Borland's Quattro Pro Tops 2.5 Million Units Shipped*, BUS. WIRE, July 1, 1992 ("Since its introduction in October 1989, Quattro Pro has won an unprecedented 42 industry awards and honors worldwide from users and product reviewers."); Lewis Peter, *Software Review, Quattro Pro 4.0; Borland International Inc.'s Spreadsheet Software*, COMPUTER SHOPPER, June 1992, at 536 ("Quattro Pro 4.0 simply shames other DOS-based spreadsheets, especially Lotus 1-2-3 r2.").

[64] *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 202, 215 (D. Mass. 1993) ("[A]lthough functional considerations may have some effect on the design of a menu tree, they do not impose any practical limitation on the possible forms of expression to a number far enough short of infinity that any way of expressing the number in English words has come into common usage"), *rev'd*, 49 F.3d 807 (1st Cir. 1995), *aff'd without opinion by equally divided court*, 516 U.S. 233 (1996).

[65] *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203, 211–14 (D. Mass. 1992), *rev'd*, 49 F.3d 807 (1st Cir. 1995), *aff'd without opinion by equally divided court*, 516 U.S. 233 (1996).

[66] Lotus Dev. Corp. v. Borland Int'l., Inc., 49 F.3d 807, 814–15 (1st Cir. 1995), *aff'd without opinion by equally divided court*, 516 U.S. 233 (1996).

granted certiorari and affirmed without opinion by an equally divided vote.[67]

Subsequent appellate decisions reached similar outcomes, although they have not fully subscribed to the First Circuit's reasoning.[68] Thus, after an inauspicious start, the federal courts implemented a balanced framework for both protecting computer software against piracy and interpreting the idea–expression dichotomy in such a way to ensure that copyright law does not extend to functional features of computer technology. Following resolution of the first API copyright war, the software engineering community believed that copyright law did not protect high–level functions, labeling conventions, and APIs.[69] Software copyright litigation subsided, and there were no new major API copyright judicial decisions until *Oracle v. Google* more than a decade later.

B. ROOTS OF THE *ORACLE V. GOOGLE* LITIGATION

The Java platform emerged with great fanfare in the mid–1990s. Just as the software copyright battles subsided, the open source movement was gaining traction, and the Internet was opening for business. The rise of the open source movement and the emergence of the Internet brought about more open and collaborative software development strategies. Sun Microsystems, the developer of Java, embraced the open bandwagon, which

[67] Lotus Dev. Corp. v. Borland Int'l, Inc., 516 U.S. 233 (1996) (Justice Stevens recused himself from participation in consideration of the case). As a result, the First Circuit's *Lotus* decision remained the law in the First Circuit but did not bind other circuits.

[68] *See* Mitel, Inc. v. Iqtel, Inc., 124 F.3d 1366, 1373–74 (10th Cir. 1997) (holding that a computer system for automating the selection of telephone long–distance carrier and remotely activating optional telecommunications features lacked the minimal degree of creativity to qualify for copyright protection and should be denied copyright protection under the *scènes à faire* doctrine because such systems are largely dictated by external factors including compatibility requirements and industry practices; but declining to hold that menu command hierarchies are categorically excluded from copyright protection); MiTek Holdings, Inc. v. ARCE Eng'g Co., 89 F.3d 1548, 1556–57 (11th Cir. 1996) (holding that the menu and submenu command structure of a software program for designing wood trusses for the framing of building roofs was uncopyrightable under § 102(b) of the Copyright Act because it represents a process).

[69] *See* Brian Proffitt, *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 ("Historically, APIs have been regarded as not falling under copyright–the reasoning being that APIs are not creative implementations but rather statements of fact"); 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/ ("Software developers routinely treat APIs as exempt from copyright protection.").

fueled widespread adoption of Java for website development. A decade after Java's release, Google foresaw the need to develop a robust open mobile operating system. It drew heavily upon Java's widely adopted language and API packages in developing Android.

1. *The Java Platform*

The phenomenal success of the Java programming language and platform was to a large extent serendipitous. To understand Java's development, it is necessary to understand the origins of Sun Microsystems in the 1980s. Sun quickly earned a reputation for its high–end computer workstations and its quirky, innovative culture.[70] Its technology fueled Silicon Valley's meteoric rise. Sun went public in 1986 under the stock symbol SUNW for Sun Workstations (later Sun Worldwide),[71] and hit $1 billion in revenues in 1988, a record for a Silicon Valley company.[72] Thanks to its reputation for cutting–edge products and engineer–friendly culture, the company attracted a talented and eclectic group of engineers and programmers.

Sun's revenues and market value grew steadily from its founding into the mid–1990s and skyrocketed during the dot–com boom that followed.[73] Flush with venture capital investment, many start–ups wanted the best workstations and servers for their engineering and programming teams.

Sun's foray into developing a new programming language began in 1990 as a skunkworks project triggered by an effort to retain top programmers. It initially aimed at developing a new generation of software to replace Sun's C and C++ APIs and tools.[74] The project evolved into developing a computer language and handheld device to control digital

[70] *See* David Bank, *The Java Saga*, WIRED (Dec. 1, 1995), http://www.wired.com/1995/12/java-saga/ (noting that while "Sun's machines had a reputation for being too complicated, too ugly, and too nerdy for mass consumption," its leadership was willing "to loosen[] the reins on some of its most precocious [programmer] talent.").

[71] *See* John Letzing, *In 1986, Sun Led the Way for Future Tech Giants*, MARKETWATCH (Oct. 19, 2009, 10:00 AM), http://www.marketwatch.com/story/sun-was-first-to-go-public-is-first-to-disappear-2009-10-19.

[72] *See* Laura Lambert, *William Joy (1954–), Programmer; Founder of Sun Microsystems*, *in* THE INTERNET: BIOGRAPHIES 138 (Hilary W. Poole ed., 2005).

[73] *See* Letzing, *supra* note 71; Lee Devlin, *The Sun Also Sets*, K0LEE.COM (Oct. 2, 2009), http://k0lee.com/2009/10/sun-also-sets/ (tracing Sun's meteoric stock rise from 1982 to 2000 and subsequent fall).

[74] *See* Bank, *supra* note 70; Nenad Dumanovic, *After 20 Years, the Java Phenomenon Just Keeps Going*, CERTIFICATION MAG. (Aug. 19, 2015), http://certmag.com/20-years-java-phenomenon-just-keeps-going/.

consumer products (such as televisions) and computers.[75] Such a language needed to be scaled for embedded systems—computer systems with a dedicated function within other systems.[76]

The team focused on developing a distributed computing environment for set–top boxes, interactive TV, and videocassette recorders through a wireless network.[77] Such a system would require a more compact footprint and hence would have more limited functionality than general purpose computing systems. By 1993, the software (codenamed Oak) was integrated into a versatile, handheld interactive TV device. Sun was unable, however, to interest consumer electronics or cable companies.[78]

Just when the project looked doomed, Bill Joy, one of Sun's founders, saw the opportunity to adapt Oak for the nascent, but promising, World Wide Web.[79] Joy realized that Oak could be repurposed to program Web pages as opposed to consumer devices. "Java," the renamed project, aimed to develop a simple, lean, platform–independent, real–time, embeddable, multitasking programming language for Web functionality. Java had a similar syntax to the widely used C language, but was far more compact, efficient, and secure. Of perhaps greatest importance, Java enabled "write once, run anywhere" (WORA) functionality—Java applets could run on Apple, Windows, or UNIX machines. Java also enabled real–time interactivity, multimedia, and animation, which greatly enhanced the dynamism of Web pages, enabling users to interact with websites in new and exciting ways.

With the experimental new software platform reaching fruition, Sun faced a conundrum. Although Sun had promoted open standards for software interfaces,[80] this project would require free release of a software

[75] *See History of the Java Programming™ Language*, WIKIBOOKS, https://en.wikibooks.org/wiki/Java_Programming/History (last visited Oct. 8, 2017).

[76] *See* Pamela Samuelson, *Foreword*, 13 BERKELEY TECH. L.J. 809, 816 (1998); Laura McNeill Hutcheson, *The Exclusion of Embedded Software and Merely Incidental Information from the Scope of Article 2B: Proposals for New Language Based on Policy and Interpretation*, 13 BERKELEY TECH. L.J. 977, 983–94 (1998); Embedded system, WIKIPEDIA, https://en.wikipedia.org/wiki/Embedded_system (last visited Oct. 8, 2017).

[77] *See* Laura Lambert, *James Gosling (1956–), Inventor of Java*, *in* THE INTERNET: BIOGRAPHIES 132–36 (Hilary W. Poole ed., 2005).

[78] Bank, *supra* note 70; Lambert, *supra* note 77, at 133–34.

[79] *See* Lambert, *supra* note 72, at 138.

[80] Sun Microsystems has been the leading member of the American Committee for Interoperable Systems (ACIS), an early lobbying organization advocating open platforms. *See* JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL: INTELLECTUAL

implementation—i.e., the full program. Marc Andreessen, the University of Illinois wunderkind who created the pioneering Mosaic web browser,[81] had released Mosaic for free for noncommercial use, but major companies were not yet in the business of giving away source code.[82] Many in the industry coveted source code as the crown jewel of high technology businesses and were loath to share it.[83]

Eric Schmidt, Sun's Chief Technology Officer who had assured the Java development team that they would be insulated from the business managers, sat at the center of an impending corporate storm. As he would later describe:

> [t]he conversation that never took place, but that I could feel all around me, was, "Eric, you are violating every principle in the company. You are taking our technology and giving it away to Microsoft and every one of our competitors. How are you going to make money?" . . . What I really believed was that Java could create an architectural franchise. The quickest way was through volume and the quickest way to volume was through the Internet.[84]

Sun invited a select group of programmers to test Java secretly in December 1994.[85] The test revealed that the WORA functionality was a game changer and word of Java's capabilities spread like wildfire throughout the programmer community.[86] Sun officially launched Java in January 1995. The business strategy epiphany came when Marc Andreessen gushed to the SAN JOSE MERCURY NEWS that "[w]hat these guys are doing is undeniably, absolutely new. It's great stuff. There's so much stuff people

PROPERTY AND INTEROPERABILITY IN THE GLOBAL SOFTWARE INDUSTRY 308 (1995) (noting that Peter Choy, who headed ACIS, worked for Sun).

[81] *See* Robert M. Yeh, *The Public Paid for the Invention: Who Owns It?*, 27 BERKELEY TECH. L.J. 453, 492 n. 235 (2012).

[82] *See* Michael Calore, *April 22, 1993: Mosaic Browser Lights Up Web with Color, Creativity*, WIRED (Apr. 22, 2010, 12:00 AM), https://www.wired.com/2010/04/0422mosaic-web-browser/.

[83] *See* Eugene A. Feher & Dmitriy S. Andreyev, *Source Code Discovery in Patent Litigation*, LAW360 (Apr. 30, 2008), http://www.law360.com/articles/54750/source-code-discovery-in-patent-litigation (noting that "most companies consider their source code to be highly confidential and part of the 'crown jewels' of the company" and that "[s]ource code frequently contains secret proprietary algorithms that provide a vital competitive advantage").

[84] *See* Bank, *supra* note 70.

[85] *See id.*; Lambert, *supra* note 72, at 139.

[86] *See* Bank, *supra* note 70 (reporting that release of early versions of Java in December 1994 "unleashed stratospheric expectations").

want to do over the network that they haven't had the software to do. These guys are really pushing the envelope."[87]

Having already released Java to a select programmer audience, Sun decided to focus on establishing Java as the standard language for web development and figure out how to make money later. It followed the "'profitless' approach to building market share" that Netscape had employed in giving away its Navigator browser.[88]

Due in part to the robust performance of its hardware divisions,[89] Sun could afford to take risks with the revenue side of its software business. Its larger concern, as manifest in the years ahead, was to prevent Microsoft from dominating the emerging Internet marketplace as it had dominated desktop computing software.[90] The WORA approach could be a game changer across the software competition landscape.[91]

In May 1995, Netscape licensed Java as part of its market–leading browser, Navigator.[92] Although Sun authorized Netscape's use for a pittance,[93] it foresaw that this move would rapidly diffuse Java across the programming community and the Web. Sun also provided Java for free to noncommercial users.[94] Java's ability to transform static web pages into engaging, animated, interactive websites revolutionized web design in a matter of months.[95]

Sun actively disseminated Java through low–cost licensing while seeking to prevent fragmentation of the Java platform.[96] Sun's strategy

[87] *See* David Bank, *Why Sun Thinks Hot Java Will Give You a Lift New Software Designed to Make World Wide Web's 'Home Pages' More Useful; And Spur Computer Sales*, SAN JOSE MERCURY NEWS, Mar. 23, 1995, at 1A; Bank, *supra* note 70 (quoting Kim Polese, Java's senior product manager: "That quote was a blessing from the god of the Internet.").

[88] *See* Bank, *supra* note 70.

[89] *See id.* (reporting that Sun's annual revenues from its hardware products were expected to exceed $6 billion in 1995).

[90] *See id.* (noting Sun co–founder and CEO Scott McNealy's rivalry with Bill Gates).

[91] *See* Mark A. Lemley & David McGowan, *Could Java Change Everything? The Competitive Propriety of a Proprietary Standard*, 43 ANTITRUST BULL. 715 (1998).

[92] *See* Lambert, *supra* note 72, at 139.

[93] *See* Bank, *supra* note 70 (reporting that Netscape "paid a paltry US$750,000" to license without any per–copy charges).

[94] *See id.*

[95] *See* Lambert, *supra* note 72, at 139.

[96] Sun sued Microsoft over its efforts to undermine the WORA principle. *See* Menell, *supra* note 8; John Markoff, *Sun Sues Microsoft in Dispute Over Java*, N.Y. TIMES (Oct. 8, 1997), https://partners.nytimes.com/library/cyber/week/100897java.html. After

succeeded in establishing Java as the de facto website programming standard.[97] Its open and low licensing cost strategy, however, meant Java was "unlikely ever to become a major profit center at Sun, though any increase in Web traffic is bound to increase sales of Sun's workstations and servers."[98] The success of Sun's hardware division through the 1990s alleviated the need for Sun to earn direct revenues from its Java division.

In 1998, Sun released the Java 2 Standard Edition Platform. It contained eight API packages, three of which—java.lang, java.io, and java.util—were necessary to use the Java programming language.[99] It also established the Java Community Process (JCP), a collaborative mechanism for Java users (including many of the leading software and hardware companies) to expand and update the Java platform.[100]

Over the ensuing years, Sun rolled out updates, improvements, and extensions. Among the goals of the JCP was to bring order to the emerging, but fragmented, mobile device ecosystem. The mobile marketplace was taking off in the mid–1990s with a variety of personal digital assistants

four years of tumultuous litigation Sun and Microsoft settled their litigation in January 2001. *See* Sun Microsystems, Inc. v. Microsoft Corp., 21 F. Supp. 2d 1109 (N.D. Cal. 1998) (granting preliminary injunction enjoining Microsoft from distributing any software implementing Java), *vacated*, 188 F.3d 115 (9th Cir. 1999), *reinstating injunction*, 87 F. Supp. 992 (N.D. Cal. 2000); *see also* Stephen Shankland, *Sun, Microsoft Settle Java Suit*, CNET (Mar. 15, 2002, 5:10 AM), http://www.cnet.com/news/sun-microsoft-settle-java-suit/. Microsoft agreed to pay Sun $20 million and was permanently prohibited from using "Java compatible" trademarks on its products. Sun would later prevail in a separate antitrust and patent infringement action against Microsoft resulting in an award of over $1 billion. *See* Scarlet Pruitt & Paul Roberts, *Microsoft to Pay $700 Million for Antitrust Issues, $900 Million to Resolve Patent Dispute*, INFOWORLD (Apr. 2, 2004), http://www.infoworld.com/article/2667124/operating-systems/update--sun--microsoft-settle-suit-in-billion-dollar-pact.html.

[97] *See* Menell, *supra* note 8.

[98] *See* Bank, *supra* note 70.

[99] *See* Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1349 (Fed. Cir. 2014) [hereinafter *Oracle II*].

[100] *See* Simon Ritter, *Keeping the Community in the Java Community Process (JCP)*, SITEPOINT (Nov. 21, 2016); *Java Community Process*, WIKIPEDIA, https://en.wikipedia.org/wiki/Java_Community_Process (last visited Oct. 8, 2017). Sun also sought to have Java recognized as a de jure standard programming language for Internet through the International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC). Sun ultimately withdrew its application as opposition mounted to a formal public standard largely controlled by a single company. *See* Lemley & McGowan, *supra* note 91, at 755.

(PDAs), cell phones, and other consumer devices. In 1998 and 1999, Sun worked with JCP members to develop the Java 2 Micro Edition (J2ME).[101]

Sun's hardware sales collapsed when the dot–com bubble burst in early 2000 as many of the companies that had ordered Sun hardware went bankrupt. Sun's stock went into freefall. As the technology sector recovered in 2004, advanced microcomputers displaced demand for higher–end Sun workstations. Sun canceled major processor projects, closed one of its two major factories, and initiated a series of layoffs.[102]

Sun came to see Java and software as its future. To expand Java's reach, Sun licensed Java Standard Edition, Enterprise Edition, and Micro Edition under the GNU GPLv2 in 2006.[103] Symbolizing its shift in direction, Sun changed its NASDAQ ticker in August 2007 from SUNW to JAVA.[104]

Sun's struggles continued, however, resulting in further deep losses during the 2008 financial crisis. Its market value fell 80% between November 2007 and November 2008, resulting in further layoffs.[105]

2. *The Android Platform*

Drawing on the Navigator and Java strategy, Google focused on widespread adoption rather than revenue for its eponymous search engine. It offered free access. As the technology press recognized its "uncanny knack for returning extremely relevant results,"[106] Google amassed loyal users and separated itself from the crowded pack of search engines. Unlike Netscape and Sun, however, Google developed a robust revenue model for its free–to–users software: keyword advertising. By October 2000, just as Sun's hardware business was setting, Google's AdWords program was

[101] *See* Ritter, *supra* note 100; *J2ME Programming/The J2ME Platform*, WIKIBOOKS, https://en.wikibooks.org/wiki/J2ME_Programming/The_J2ME_Platform (last visited Oct. 8, 2017).

[102] *See* Menell, *supra* note 8.

[103] *See* Steven J. Vaughan-Nichols, *Sun to Open-Source Java under GPL*, PRACTICAL TECH. (Nov. 11, 2006), http://practical-tech.com/development/sun-to-open-source-java-under-gpl/415/. The GNU GPL license requires that software built on the open source code base be available to others on an open source basis—the so–called share–alike requirement. *See* Brian C. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 BERKELEY TECH. L.J. 443, 455 (2005).

[104] *See Sun Microsystems' New Ticker: JAVA*, L.A. TIMES (Aug. 24, 2007), http://articles.latimes.com/2007/aug/24/business/fi-wrap24.s4.

[105] *See* Devlin, *supra* note 73; Ashlee Vance, *Sun Microsystems Reports $1.7 Billion Loss and Falling Sales*, N.Y. TIMES (Oct. 30, 2008), http://www.nytimes.com/2008/10/31/technology/companies/31sun.html.

[106] *See* Don Willmot, *Top 100 Web Sites*, PC MAG., Feb. 9, 1999, at 118.

launched.[107] In August 2001, Google named Eric Schmidt, Sun's former CTO as its Chief Executive Officer. The press release touted that Schmidt had "led the development of Java, Sun's platform-independent programming technology, and defined Sun's Internet software strategy."[108]

With revenue flowing from AdWords, Google developed a series of new projects—image search, news, shopping, Gmail, maps—which reinforced and expanded its advertising business. Google went public in 2004[109] and continued to expand its reach with Google Books, YouTube, and other projects.[110]

Google's leaders foresaw mobile devices as a substantial risk to their advertising juggernaut.[111] The early smartphones, such as RIM's BlackBerry, did not make effective use of Google's advertising links.[112]

[107] *See* GOOGLE CO., *Our History in Depth*, https://www.google.com/about/company/history (last visited Oct. 8, 2017); Jemima Kiss, *Ten Years of Online Advertising with Google Adwords*, GUARDIAN (Oct. 25, 2010, 2:00 PM), https://www.theguardian.com/media/2010/oct/25/advertising-google-adwords.

[108] *See Google Names Dr. Eric Schmidt Chief Executive Officer*, GOOGLE (Aug. 6, 2001), www.googlepress.blogspot.com/2001/08/google-names-dr-eric-schmidt-chief.html.

[109] *See* John Markoff, *THE GOOGLE I.P.O.: THE OVERVIEW; Google's Sale of Its Shares Will Defy Wall St. Tradition*, N.Y. TIMES (Apr. 30, 2004), http://www.nytimes.com/2004/04/30/business/google-ipo-overview-google-s-sale-its-shares-will-defy-wall-st-tradition.html.

[110] *See* Stephen Shankland, *Google Takes Bold Action to Match Aspirations*, CNET (Aug. 15, 2011, 10:54 AM), https://www.cnet.com/news/google-takes-bold-action-to-match-aspirations/.

[111] In its 2005 10-K filing, Google identified the emerging mobile marketplace as a potential threat to its profitability:

> ***More individuals are using non-PC devices to access the Internet, and versions of our web search technology developed for these devices may not be widely adopted by users of these devices.***
>
> The number of people who access the Internet through devices other than personal computers, including mobile telephones, hand-held calendaring and e-mail assistants, and television set-top devices, has increased dramatically in the past few years. The lower resolution, functionality and memory associated with alternative devices make the use of our products and services through such devices difficult. If we are unable to attract and retain a substantial number of alternative device users to our web search services or if we are slow to develop products and technologies that are more compatible with non-PC communications devices, we will fail to capture a significant share of an increasingly important portion of the market for online services.

Google Inc., Annual Report 32 (Form 10-K) (Dec. 31, 2005).

[112] *See* FRED VOGELSTEIN, DOGFIGHT: HOW APPLE AND GOOGLE WENT TO WAR AND STARTED A REVOLUTION 53 (2013).

Developing a new mobile platform, however, posed daunting challenges. The mobile marketplace was a morass of telecommunication companies, handset makers, and software providers.[113] The telecommunications companies (telcos) jealously guarded their networks.[114] The handset makers (original equipment manufacturers (OEMs)) had divergent strategies and business models. Microsoft and Symbian were promoting proprietary mobile operating systems but without notable success. Google executives feared that Microsoft could steer consumers away from Google search if Microsoft successfully established a mobile platform.[115]

Google saw promise in Android, a startup founded in October 2003 to develop "smarter mobile devices that are more aware of [their] owner's location and preferences."[116] Android's founder, Andy Rubin, had previously developed T–Mobile's Sidekick, a compact mobile device that provided an authentic web browsing experience.[117] Rubin and Google recognized that an open, competitive strategy could potentially overcome the structural factors impeding development of a breakthrough mobile platform.[118] Google acquired Android for $50 million in July 2005 and put Rubin in charge of its new mobile division.[119]

At the first high-level Android planning meeting, the newly established Android team and Google leaders focused on three questions:

> • Which type of Open Source are we?
>
> • How do we interact with the OSS [open source software community]?
>
> • How do we Open Source our JVM [Java Virtual Machine]?[120]

[113] *See id.* at 48–50.

[114] *See* John Markoff, *I, Robot: The Man Behind the Google Phone*, N.Y. TIMES (Nov. 4, 2007), http://www.nytimes.com/2007/11/04/technology/04google.html.

[115] *See* VOGELSTEIN, *supra* note 112, at 51.

[116] *See* Ben Elgin, *Google Buys Android for Its Mobile Arsenal*, BUS. WEEK (Aug. 17, 2005), http://tech-insider.org/mobile/research/2005/0817.html.

[117] *See* John Markoff, *Where Does Google Plan to Spend $4 Billion?*, N.Y. TIMES (Aug. 22, 2005), http://www.nytimes.com/2005/08/22/technology/where-does-google-plan-tospend-4-billion.html (observing that Page and Brin wore the Sidekick all–purpose voice and data communicators on their belts several years ago and that Page had long envisioned a Google branded smart phone); VOGELSTEIN, *supra* note 112, at 52–53.

[118] *See* VOGELSTEIN, *supra* note 112, at 49.

[119] *See* Markoff, *supra* note 117.

[120] *See* Trial Ex. 1, Android GPS [Google Product Strategy]: Key strategic decisions around Open Source at 2, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (filed July 26, 2005) [hereinafter Trial Exhibit 1].

The Android team planned to use a permissive open source license that merely required licensees to maintain compatibility with Google APIs.[121] Several factors made Java a critical part of their plan: the carriers required it; Microsoft would never pursue it; Java had well–developed, tested tools; Java assured third party app developers that the platform would remain available; a large and growing pool of developers knew Java; and handset makers could cheaply license Java from Sun.[122]

Sun's Java 2 Mobile Edition, which was widely used on feature phones,[123] would not be adequate for the Android platform for several reasons. First, Google sought to design a new platform optimized for the small chips on handsets and add new functionalities.[124] It aimed to use some of the Java API packages and develop others. Second, Google sought to use a less restrictive licensing model than the GNU GPL in order to promote robust downstream innovation and competition.[125] The GPL's viral "share and share alike" provision would prevent handset makers and telephone companies from commercializing proprietary extensions on top of the base platform.[126] Furthermore, these vendors would not want to share that technology under the viral licensing model.[127]

The Android team believed that they could achieve their goals by negotiating the first open source Java 2 Platform, Micro Edition license with Sun.[128] The preliminary negotiations went well, and both sides believed

[121] *See id.*

[122] *See id.* at 8.

[123] The term "feature phone" characterizes low–end mobile phones with limited capability–principally voice and text messaging with basic multimedia and rudimentary internet access. They have relatively small screens. *See* Nicole Lee, *The 411: Feature Phones vs. Smartphones*, CNET (Mar. 1, 2010, 5:14 PM) https://www.cnet.com/news/the-411-feature-phones-vs-smartphones/.

[124] *See* Trial Exhibit 1, *supra* note 120, at 2.

[125] *See id.*

[126] *See id.*

[127] *See* Trial Ex. 230, email from Andy Rubin to Bob Lee (Aug. 11, 2007), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) ("The problem with GPL in embedded systems is that it's viral, and there is no way (for example) OEMs or Carriers to differentiate by adding proprietary works. We are building a platform where the entire purpose is to let people differentiate on top of it.").

[128] *See* Trial Exhibit 1, *supra* note 120, at 2. The memo noted that Tim Lindholm, a former Sun Microsystems engineer who was involved with Java, and who then worked for Google, would lead the negotiation. *See id.* at 9; John Letzing, *Who Is Tim Lindholm? Google's CEO is Wondering That Too*, WALL ST. J. (Apr. 18, 2012), http://blogs.wsj.com/digits/2012/04/18/who-is-tim-lindholm-googles-ceo-is-wondering-that-too/.

they could reach an agreement.[129] The negotiations unraveled, however, over Google's unwillingness to agree to make Android fully compatible with the Java platform.[130] Sun demanded strict adherence to the WORA principle, which was a deal–breaker for Google.

Google pushed ahead with its selective use of Java API packages by independently implementing the functional specifications in a clean–room environment.[131] Using the Java language would not be a problem as it had long been freely available. But the Android team also wanted to use selected Java API packages from the Java™ Standard Edition and develop its own virtual machine. If the Java programming language is analogized to the letters, words, and syntax of the English language, the API implementations can roughly be characterized as paragraphs or chapters within a book. Copying the full API implementations—involving large chunks of source code—would run afoul of copyright law. Android could achieve its goals by emulating the API functionality with independently written implementing code. And by avoiding restrictive licensing terms with Sun, Google could blaze its own trail free of Sun's interference.[132]

[129] *See* Trial Ex. 13, email from Brian Swetland to Mathias Agopian (Jan. 2, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA); Doc. 398-10, email from Andy Rubin to Sergey Brin, (Jan. 13, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA), http://www.fosspatents.com/2011/09/sun-proposed-red-hat-style-android.html; Trial Ex. 16, email from Scott McNealy to Vineet Gupta (Feb. 9, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA); Trial Ex. 2372, email from Vineet Gupta to Andy Rubin (May. 8, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA).

[130] *See* Trial Ex. 214, email from Eric Schmidt to Andy Rubin (May 14, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA); Trial Ex. 215, email from Chris Desalvo to Andy Rubin (June 1, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA); VOGELSTEIN, *supra* note 112, at 57 (reporting that Sun would not agree to forking of its platform); *see also* Trial Ex. 230, email from Andy Rubin to Bob Lee (Aug. 11, 2007), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (explaining Sun's profit motivation for choosing GPL for Java ME: "Sun chose GPL . . . so that companies would need to come back to them and take a direct license and pay royalties," and noting that Google "negotiated 9 months with Sun and decided to walk away after they threatened to sue us over patent violations.").

[131] *See* Trial Ex. 215, email from Chris Desalvo to Andy Rubin (June 1, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) ("With talks with Sun broken off where does that leave us regarding Java class libraries? Ours are half-ass at best. We need another half of an ass.").

[132] *See* Trial Ex. 18, email from Andy Rubin to Greg Stein (Mar. 24, 2006), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (expressing consternation at Sun's licensing model: "Ha, wish them luck. Java.lang api's

Google recognized that this path risked copyright and patent infringement. The copyright issue turned on whether and to what extent copyright law protected the function labels and structure, sequence, and organization (SSO) of Java APIs. As a result of the Supreme Court's deadlock, the *Lotus v. Borland* decision, which cleared the way for copying of function labels, strictly governed only the First Circuit. The Second Circuit's *Altai* decision and the Ninth Circuit's *Apple* decision exposed the weakness of the Third Circuit's superficial analysis of SSO in *Whelan*. Furthermore, the *Altai* decision and the Ninth Circuit's *Sega* decision clearly viewed achieving interoperability with another computer interface through a different implementation to be fair game, but Android was aiming for something other than end–user interoperability. It wanted to pick and choose among interface elements in building a new platform—an optimized interface for a different consumer product.

Over the next two years, the Android team independently implemented 37 of the 166 Java API packages in the Java™ Standard Edition[133] and developed an independent virtual machine ("Dalvik"). In this way, the Android operating system emulated the functionality of known and tested APIs that fit the Android team's constrained design parameters. Android's use of the same function labels as Java would enable millions of Java programmers to quickly master Android app development. Although Android apps would not be fully interoperable with Java, they would be similar and better optimized to the constraints of mobile devices.[134] This clean–room effort added substantially more time and cost to Android's development, but avoided literal copying of the Java API implementation code.[135]

With the breakthrough success of the Apple iPhone in 2007, Google came to see Android as critical to its business strategy.[136] The iPhone propelled Apple into a dominant position in the emerging smartphone marketplace. Google feared that Apple could rule mobile technology in

are copyrighted. And Sun gets to say who they license the tck [Technology Compatibility Kit used to ensure Java compatibility] to, and forces you to take the 'shared part' which taints any clean room implementation.").

[133] *See* Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974, 977 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) [hereinafter *Oracle I*].

[134] *See* Stephen Shankland, *Google Carves an Android Path Through Open-source World*, CNET (May 22, 2008), http://www.cnet.com/news/google-carves-an-android-path-through-open-source-world/.

[135] *See* VOGELSTEIN, *supra* note 112, at 57.

[136] *See id.* at 129–30.

much the same way that Microsoft had ruled the desktop market, thereby threatening Google's strength in search and other Internet services. In response, Google allocated more resources to the Android project.[137] The Android team found it far easier to negotiate partnerships with the many telcos and handset manufacturers marginalized by Apple's decision to produce its own device and license the iPhone exclusively with AT&T.[138]

Google began the rollout of the Android platform in early November 2007.[139] On November 5th, Google unveiled the Open Handset Alliance, a consortium of handset makers, application developers, telcos, and components manufacturers (such as chip makers), in conjunction with the outlines of the Android platform.[140] Andy Rubin explained that Android's software was based on the Linux operating system and Sun's Java language, which would enable programmers to easily develop applications that connect to independent Web services.[141]

Jonathan Schwartz, Sun's CEO, publicly applauded Google's use of Java, proclaiming that Google had "strapped another set of rockets to the [Java] community's momentum—and to the vision-defining opportunity across our (and other) planets."[142] Privately, Sun feared that Android's use of Java would undermine its WORA paradigm and its mission to establish Java ME as the leading mobile platform and a significant revenue generator.[143]

[137] *See* Trial Ex. 433, Android GPS Meeting Notes (July 17, 2007), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA); VOGELSTEIN, *supra* note 112, at 83–84.

[138] *See* VOGELSTEIN, *supra* note 112, at 119–21.

[139] *See Industry Leaders Announce Open Platform for Mobile Devices: Group Pledges to Unleash Innovation for Mobile Users Worldwide*, OPEN HANDSET ALLIANCE (Nov. 5, 2007), http://www.openhandsetalliance.com/press_110507.html; Miguel Helft & John Markoff, *Google Enters the Wireless World*, N.Y. TIMES (Nov. 5, 2007), http://www.nytimes.com/2007/11/05/technology/05cnd-gphone.html; Saul Hansell, *The Gphone: So Open It Could Be Closed*, N.Y. TIMES: BITS BLOG (Nov. 5, 2007), http://bits.blogs.nytimes.com/2007/11/05/the-gphone-so-open-it-could-be-closed/.

[140] *See* Erick Schonfeld, *Breaking: Google Announces Android and Open Handset Alliance*, TECHCRUNCH (Nov. 5, 2007), https://techcrunch.com/2007/11/05/breaking-google-announces-android-and-open-handset-alliance/.

[141] *See* Helft & Markoff, *supra* note 139.

[142] *See* Jay Greene, *Scoop: Oracle Scrubs Site of Embarrassing Java Blog*, CNET (July 22, 2011, 10:09 PM), https://www.cnet.com/uk/news/scoop-oracle-scrubs-site-of-embarrassing-java-blog/.

[143] *See* Trial Ex. 565, email thread from Vineet Gupta (Sep. 24, 2007), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA). In an "off-the-record" communication with a New York Times reporter one day after the Android announcement, Schwartz sniped about Google's opposition to Sun's plan to open

The following week, Google released the "open source" Android Software Development Kit (SDK), which enabled companies to build their own smartphones.[144] Google did not, however, release the Android source code, indicating that it would not be available until the first Android phones went on sale in late 2008.[145] Nor did Google release its own branded phone, although it left that option open.

Based on the Android SDK, Sun and other industry observers could see that Google was diverging from the Java standard platform and the Java Community Process.[146] Google deflected suggestions that Android fragmented Java by focusing attention on how the Open Handset Alliance provides a more responsive, less restrictive, open platform for mobile devices.[147] Sun and Google continued to monitor each other's activities warily as Android products moved into the marketplace in 2008 and

source Java. *See* Trial Ex. 2371, email from Jonathan Schwartz to John Markoff (Nov. 6, 2007), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA), http://www.fosspatents.com/2012/04/former-sun-chief-about-google-immune-to.html.

[144] *See* Steve Horowitz, *Calling All Developers: $10M Android Challenge*, GOOGLE OFFICIAL BLOG (Nov. 12, 2007), https://googleblog.blogspot.com/2007/11/calling-all-developers-10m-android.html ("Today, the team is releasing an early look at the Android SDK for developers interested in building applications for Android. To get things rolling, we've also announced the Android Developer Challenge, which provides $10 million in awards for developers who build great applications for Android.").

[145] *See* Peter Judge, *Google Android on the Defensive: Google Defends Decision to Use its Own Flavor of Java in the Android SDK Rather Than Support the Popular C++*, INFOWORLD (Nov. 15, 2007), http://www.infoworld.com/article/2651252/networking/google-android-on-the-defensive.html.

[146] *See* Stephen Shankland, *Sun's Worried That Google Android Could Fracture Java*, CNET (Nov. 14, 2007), http://www.cnet.com/news/suns-worried-that-google-android-could-fracture-java/ ("Painful flashbacks are beginning to torment those of us who lived through the Java wars between Sun Microsystems and Microsoft that began 10 years ago. Earlier this week, Google released programming tools for its Android mobile-phone software project that shun the existing Java standard-setting process in favor of a Google-specific variety. Sun responded on Wednesday by expressing concern that Google's Android project could fragment Java into incompatible versions."); *see also* Stephen Shankland, *Google's Android Parts Ways with Java Industry Group*, CNET (Nov. 13, 2007), http://www.cnet.com/news/googles-android-parts-ways-with-java-industry-group/.

[147] *See* Stephen Shankland, *Sun's Worried That Google Android Could Fracture*, CNET (Nov. 14, 2007), http://www.cnet.com/news/suns-worried-that-google-android-could-fracture-java/ (quoting a Google press statement: "Google and the other members of the Open Handset Alliance are working to help solve fragmentation and supporting the developer community by creating Android, a mobile platform that responds to the needs of the developers, has the backing of industry leaders, and will be available as open source under a nonrestrictive license.").

2009,[148] a period in which Apple's iPhone was ascendant. Leaders at both companies occasionally broached licensing and collaboration,[149] but a gulf remained.[150] Sun refrained from blocking Android through legal action.

The marketplace resolved the fate of the two companies. Rubin's vision proved prescient: "When you have multiple [handset makers] building multiple products in multiple product categories, it's just a matter of time before sales of Android phones exceed the sales of proprietary systems like Apple's and R.I.M.'s."[151] After a gradual start, Android took the global smartphone operating systems market by storm, surpassing the Apple iOS

[148] *See* Trial Ex. 2070, email from Vineet Gupta to Jonathan Schwartz (Oct. 23, 2008), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (indicating that Google's Android "proposal more than likely is going to be about buying out Java"); Trial Ex. 29, email from Rubin to Dick Wall (Mar. 24, 2008), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (warning Google representatives not to demonstrate Android features to Sun employees or lawyers at JavaOne convention), Trial Ex. 326, email from Dave Sobota to Tim Lindholm (Feb. 19, 2009), Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (raising the question of who will own Java if Sun collapses and suggesting Google could buy the patent and copyright rights as a way of making "[o]ur Java lawsuits go away"); Trial Ex. 1029, email from Tim Lindholm to Dan Bornstein (Apr. 29, 2009), 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (recommending avoiding interaction with Sun to avoid "inadvertently stir[ring] anything up for Android").

[149] *See* Trial Ex. 1002, email thread from Tim Lindholm to Andy Rubin (Nov. 24, 2008), 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (discussing recent efforts by Sun to "certify Android through the Java process and become licensees of Java"); Trial Ex. 3466, email from Eric Schmidt to Jonathan Schwartz (Mar. 31, 2008), 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA) (Re: update on android licensing; "We are happy to have our team meet with anyone at Sun who would like more information or who has ideas for us"; calling attention to an explanation of why Google chose to distribute Android to the public using the Apache v2 license); *see also* Ryan Paul, *Why Google Chose the Apache Software License over GPLv2 for Android*, ARS TECHNICA (Nov. 6, 2007), http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/ (linked in Eric Schmidt's March 31, 2008 email to Jonathan Schwartz).

[150] Sun had proposed to license Java to Google for $60 million over three years plus an additional amount of up to $25 million per year in revenue sharing. *See* Doc. 182, Letter from Scott Weingaertner, Counsel, Google, to Judge William Alsup at 5 (June 6, 2011) 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA), https://www.scribd.com/document/58133136/Oracle-Google-Damages-June-6-Precis-Unredacted. It is unclear whether that offer would have afforded Google the flexibility and independence in developing Android that it sought.

[151] *See* Brad Stone, *Google's Andy Rubin on Everything Android*, N.Y. TIMES: BITS BLOG (Apr. 27, 2010), http://bits.blogs.nytimes.com/2010/04/27/googles-andy-rubin-on-everything-android/ (internal quotation marks omitted).

market share by mid–2010 and leaving Java ME, RIM, Microsoft, and Symbian in the dust.[152]

With its hardware business in decline, software acquisitions sputtering,[153] and ability to monetize Java diminished, Sun Microsystems' ability to survive as an independent company came into question.[154] Oracle Corporation, one of the strongest software companies that had built many of its products on the Java platform, swooped in.[155]

Oracle's acquisition of Sun brought legal action against Google into play. Notwithstanding consternation over Android's "unofficial," nonstandard, and incomplete Java implementation,[156] suing Google would have gone against Sun's long–standing cultural norms about open technology and evangelism within the industry.[157] Moreover, Sun could ill afford a prolonged litigation battle or the risk to Sun's reputation with other technology companies. Google was well–positioned financially and legally to put up a stiff defense. Sun's business was struggling, and Wall Street and

[152] *See Global Mobile OS Market Share in Sales to End Users From 1st Quarter 2009 to 1st Quarter 2017*, STATISTA (2017), http://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/.

[153] Sun had purchased StorageTek, a storage vendor, for $4.1 billion in 2005 and MySQL, a relational database company, for $1 billion in 2008. *See* Jon Brodkin, *The Downfall of Sun Microsystems*, NETWORKWORLD (Apr. 24, 2009), http://www.networkworld.com/article/2268096/servers/the-downfall-of-sun-microsystems.html.

[154] *Id.*

[155] *See* Patrick Thibodeau & Elizabeth Montalbano, *Update: Oracle Buying Sun in $7.4B Deal*, COMPUTERWORLD (Apr. 20, 2009), http://www.computerworld.com/article/2523479/data-center/update--oracle-buying-sun-in--7-4b-deal.html.

[156] *See* Dan Farber, *Java Creator James Gosling: 'Google Totally Slimed Sun'*, CNET (Apr. 30, 2012), http://www.cnet.com/news/java-creator-james-gosling-google-totally-slimed-sun/ (quoting Gosling stating that Sun was "wronged" by Google and citing Sun's objections to Android's "very weak notions of interoperability" with Java); Gavin Clarke, *Ellison Wrestles Google to Strangle 'Unofficial' Java*, REGISTER (Aug. 13, 2010), (referring to Android as an "unofficial" Java software platform), https://www.theregister.co.uk/2010/08/13/oracle_google_java_prosecution/; Joe Mullin, *Sun's Jonathan Schwartz at Trial: Java Was Free, Android Had No Licensing Problem*, ARS TECHNICA (May 11, 2016), http://arstechnica.com/tech-policy/2016/05/suns-jonathan-schwartz-at-trial-java-was-free-android-had-no-licensing-problem/ (quoting former Sun CEO expressing annoyance at Google's refusal to work out a license with Sun).

[157] *See* Brad Feld, *Oracle's Java API Suit Against Google—Five Years Later*, FELDTHOUGHTS (June 29, 2015), http://www.feld.com/archives/2015/06/oracles-java-api-suit-google-five-years-later.html (observing that "[f]iling patent suits was never in Sun's genetic code" (quoting James Gosling, *The Shit Finally Hits the Fan....*, JAVAVIRTUALMACHINE.NET (Aug. 13, 2010), http://news.java-virtual-machine.net/6018.html)); Mullin, *supra* note 156.

potential suitors would likely have seen such a lawsuit as a sign of desperation and a distraction from Sun's business goals.

Oracle's acquisition of Sun Microsystems dramatically altered the Java enforcement equation. Larry Ellison, Oracle's cofounder and CEO, had a reputation for brash business tactics. Whereas Sun's leadership had embraced open technology with religious fervor, Oracle's approach had been strategic. Unlike Sun, Oracle possessed the financial strength and diversified business strategy to pursue high stakes litigation. It had done well in recent years pursuing corporate takeovers and copyright litigation against SAP.[158]

In announcing the Sun acquisition, Ellison characterized Java as "the single most important software asset we have ever acquired" and touted Oracle's Java–based, middleware business, bolstered first by its BEA Systems acquisition[159] and purchase of Sun, as being "on track to become as large as Oracle's flagship database business."[160] Oracle would need to reposition Java's licensing business to achieve that goal. Oracle's leadership team sought to pursue a far more aggressive Java licensing strategy. It believed that Sun products could bring in $1.5 billion in operating profits in the first year following the acquisition.[161]

Oracle completed its acquisition of Sun in early 2010.[162] Oracle immediately approached Google about its use of Java in the Android

[158] *See* Jim Henschen, *Oracle Lawsuit Against SAP Settled at Law*, INFORMATIONWEEK (Nov. 14, 2016, 8:50 AM), http://www.informationweek.com/cloud/software-as-a-service/oracle-lawsuit-against-sap-settled-at-last/d/d-id/1317483; Verne F. Kopytoff, *SAP Ordered to Pay Oracle $1.3 Billion*, N.Y. TIMES (Nov. 23, 2010), http://www.nytimes.com/2010/11/24/business/24oracle.html; Danny Diegal, *Oracle Takes Reduced $357M Award in SAP Copyright Suit*, LAW360 (Nov. 13, 2016, 9:42 PM), https://www.law360.com/articles/596188/oracle-takes-reduced-357m-award-in-sap-copyright-suit; Steve Lohr & Laurie J. Flynn, *Oracle to Acquire PeopleSoft for $10.3 Billion, Ending Bitter Fight*, N.Y. TIMES (Dec. 14, 2004), http://www.nytimes.com/2004/12/14/technology/oracle-to-acquire-peoplesoft-for-103-billion-ending-bitter-fight.html.

[159] *See* Larry Dugan, *Surprise! Oracle Buys BEA Systems*, ZDNET (Jan. 16, 2008), http://www.zdnet.com/article/surprise-oracle-buys-bea-systems/. BEA Systems specialized in enterprise infrastructure software products. *See id.*

[160] *See* Patrick Thibodeau & Elizabeth Montalbano, *Update: Oracle Buying Sun in $7.4B Deal*, COMPUTERWORLD (Apr. 20, 2009), http://www.computerworld.com/article/2523479/data-center/update--oracle-buying-sun-in--7-4b-deal.html.

[161] *See* Brodkin, *supra* note 153.

[162] Antitrust authorities in the U.S. and Europe delayed the acquisition out of concern that Oracle—the leading relational database vendor—was acquiring a promising competing business (MySQL). *See* James Kanter, *New Snag for Oracle in Sun Deal*, N.Y.

platform. Google considered alternatives to Java,[163] but ultimately stood its ground due to the lack of adequate workarounds. For Oracle, the prospect of spending millions of dollars on attorneys' fees for even a modest possibility of sharing in the large and growing Android marketplace was a plausible, if not attractive, business proposition. Moreover, it could quickly establish Oracle as a key player in the lucrative, strategically important, and rapidly growing mobile operating system marketplace. Delay would only enhance Google's laches and equitable–estoppel defenses.

C. THE *ORACLE V. GOOGLE* LITIGATION

After six months of negotiations with Google, Oracle filed a broadside salvo alleging Android infringed Java–related patents and copyrights in the Northern District of California.[164] With billions of dollars and control of two of the most important software platforms at stake, the parties spared no expense in litigating the case over the next six years.

The case was assigned to Judge William Alsup, a respected jurist who was unafraid to grapple with complex technologies.[165] Judge Alsup actively managed the case, pushing the parties to settle the dispute or move to trial quickly. He pressured Oracle to streamline its patent allegations[166] and

TIMES (Sept. 3, 2009), http://www.nytimes.com/2009/09/04/technology/companies/04oracle.html.

[163] *See* Florian Mueller, *Google's Five Failed Attempts to Give Confidential Status to 'Damning' E-mail in Oracle Case*, FOSS PATENTS (Nov. 9, 2011), http://www.fosspatents.com/2011/11/googles-five-failed-attempts-to-give.html (noting that Page and Brin had asked engineers "to investigate what technical alternatives exist to Java for Android and Chrome. We have been over a bunch of these, and think they all suck. We conclude that we need to negotiate a license for Java under the terms we need." (quoting Trial Ex. 10, email from Tim Lindholm to Andy Rubin et al. (Aug. 6, 2010), Oracle Am., Inc. v. Google Inc. 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA))).

[164] *See* Complaint for Patent and Copyright Infringement, Oracle Am., Inc. v. Google, Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA), 2010 WL 11221486.

[165] *See* Dan Farber, *Judge William Alsup: Master of the Court and Java*, CNET (May 31, 2012), http://www.cnet.com/news/judge-william-alsup-master-of-the-court-and-java/.

[166] *See* Florian Mueller, *Oracle-Google Trial to Start on April 16, 2012*, FOSS PATENTS (Mar. 13, 2012), http://www.fosspatents.com/2012/03/oracle-google-trial-to-start-on-april.html; Florian Mueller, *Oracle Offers Withdrawal of Three More Patents in Exchange for Spring Trial Against Google*, FOSS PATENTS (Mar. 9, 2012), http://www.fosspatents.com/2012/03/oracle-offers-withdrawal-of-three-more.html; Florian Mueller, *Pressure Mounting on Oracle to Drop Patent Claims Against Google and Focus on Copyright*, FOSS PATENTS (Mar. 5, 2012), http://www.fosspatents.com/2012/03/pressure-mounting-on-oracle-to-drop.html.

rejected Google's summary judgment motion, which asserted that the API packages were uncopyrightable.[167] While agreeing with Google that "the names of the Java language API files, packages, classes, and methods are not protectable as a matter of law"[168] under the copyright doctrine denying protection for names and short phrases,[169] the court nonetheless rejected Google's broader argument that API declarations (beyond short phrases) and documentation are unprotectable under the *scènes à faire*, merger, or methods of operation (§ 102(b)) doctrines.

1. *The 2012 Trial*

Judge Alsup structured the trial in three phases: (I) copyright infringement claims; (II) patent infringement claims; and (III) all remaining issues, including damages and willfulness, if necessary.[170] As the case wended its way toward trial, the core copyright allegations were boiled down to: (a) "12 Android files of source code (copied from 11 Java files), including rangeCheck"; (b) "Plain English descriptions in the user manual, sometimes called the API 'specifications'"; (c) "37 APIs but only as to their specific selection, structure, and organization, it being conceded that the implementing code is different"; and (d) "Android's entire source code and object code as derivative works of the 37 Java APIs."[171] The parties agreed that Judge Alsup would decide the copyrightability of the Java APIs and that the jury would decide copyright infringement, fair use, and whether any copying was de minimis.[172] Thus, a jury would not hear the most salient copyright issue the Oracle–Google litigation raised—the copyrightability of APIs.

[167] *See* Oracle Am., Inc. v. Google Inc., 810 F. Supp. 2d 1002, 1005 (N.D. Cal. 2011).

[168] *Id.* at 1009–10.

[169] *See* Material Not Subject to Copyright, 37 C.F.R. 202.1(a) (2016) (regulation denying copyright registration for "[w]ords and short phrases such as names, titles, and slogans"); *see also* Planesi v. Peters, No. 04-16936, 2005 WL 1939885, *1 (9th Cir. Aug. 15, 2005); Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1524 n.7 (9th Cir. 1992) ("Sega's security code is of such *de minimis* length that it is probably unprotected under the words and short phrases doctrine.").

[170] *See* Oracle Am., Inc. v. Google Inc., 3:10-cv-03561-WHA, 2012 WL 1189898, (N.D. Cal. Jan. 4, 2012).

[171] *See* Request for Statement of Issues Re: Copyright at 1–2, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA), https://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/854.

[172] *Oracle I*, 872 F. Supp. 2d at 975.

As a result of Judge Alsup's decision to reserve the API copyrightability question to himself, the jury's infringement verdict was largely a foregone conclusion. Judge Alsup instructed the jury that Oracle's Java–related copyrights "cover the structure, sequence and organization [SSO] of the compilable code"[173] and that Google "agrees that the structure, sequence and organization of the 37 accused API packages in Android is substantially the same as the structure, sequence and organization of the corresponding 37 API packages in Java."[174] Judge Alsup further instructed the jury that "[w]hile individual names are not protectable on a standalone basis, names must necessarily be used as part of the structure, sequence, and organization and are to that extent protectable by copyright."[175]

Oracle's principal copyright infringement argument boiled down to showing the jury a side–by–side comparison of Java and Android source code. Beyond its motion seeking a determination that the Java APIs are not copyrightable,[176] Google's principal path to a trial victory was that the jury would find that Android's use of Java was permissible under the fair use doctrine.

Oracle secured a partial victory in the copyright phase of the trial.[177] While concluding that Android infringed the 37 Java API packages in question taken as a group,[178] the jury found that Google did not infringe

[173] *See* Doc. 1018, Final Charge to the Jury (Phase One) and Special Verdict Form at 8, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed Apr. 30, 2012), https://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1018.

[174] *See id*. at 10.

[175] *See id*. at 20.

[176] *See* Doc. 984, Google's Motion for Judgment as a Matter of Law on Sections of Count VIII of Oracle's Amended Complaint, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA), 2012 WL 3992644.

[177] *See* Doc. 1089, Special Verdict Form, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed May 7, 2012),https://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1089; Joe Mullin, *Google Guilty of Infringement in Oracle Trial*, ARS TECHNICA (May 7, 2012), http://arstechnica.com/tech-policy/2012/05/jury-rules-google-violated-copyright-law-google-moves-for-mistrial/.

[178] *See* Doc. 1089, Special Verdict Form at 1, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed May 7, 2012), https://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1089.

Java documentation[179] and that the copying of eight of the nine specific source code files at issue was de minimis.[180] The jury hung on whether Google's infringement of the Java API SSO constituted fair use.[181] The jury split on the special interrogatories relating to Google's equitable–estoppel defense. It found Sun/Oracle had engaged in conduct that they knew or should have known would lead Google to believe reasonably that it would not need a license to use the Java API SSO. Google nevertheless had not proven that it relied on Sun's conduct.[182]

The patent phase of the trial commenced shortly after the jury rendered its copyright verdict. The same jury ruled that Google did not infringe the eight asserted claims of the two patents at issue.[183] Therefore, the need for the third phase of the trial hinged on Judge Alsup's resolution of the post–trial copyright motions.

Shortly after the patent phase of the trial ended, Judge Alsup issued a detailed opinion holding that the Java APIs were not copyrightable,[184] resulting in dismissal of the case. Although Judge Alsup cautioned that the ruling did not hold that "Java API packages are free for all to use without license" or that "the structure, sequence, and organization of all computer programs may be stolen," the court held "on the specific facts of this case [that] the particular elements replicated by Google were free for all to use under the Copyright Act."[185]

[179] *See* Doc. 1018, Final Charge to the Jury (Phase One) and Special Verdict Form at 12, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed Apr. 30, 2012), https://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1018.

[180] *See* Doc. 1089, Special Verdict Form at 2, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed May 7, 2012), https://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1089.

[181] *See id.* at 1.

[182] *See id.* at 3.

[183] *See* Doc. 1190, Special Verdict Form, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed May 23, 2012), https://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1190; Josh Lowensohn, *Jury Verdict: Android Doesn't Infringe Oracle's Patents*, CNET (May 23, 2012, 11:06 AM), http://www.cnet.com/news/jury-verdict-android-doesnt-infringe-oracles-patents/.

[184] *See Oracle I*, 872 F. Supp. 2d 974 (N.D. Cal. 2012). In a pyrrhic victory for Oracle, Judge Alsup granted judgment as a matter of law holding that Google's copying of the eight test files that the jury deemed de minimis were infringing. *See* Oracle Am., Inc. v. Google Inc., No. C 10–3561, 2012 U.S. Dist. LEXIS 66417 (N.D. Cal. May 11, 2012).

[185] *Oracle I*, 872 F. Supp. 2d at 1002.

Judge Alsup grounded his decision in the particular and distinctive functional attributes of the 37 Java APIs and the fact that Google wrote its own implementing code.[186] The principal copying concerned the lines of declaring code, which are necessary to operate the particular methods of the APIs. As Judge Alsup explained,

> Significantly, 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.[187]

While acknowledging that the overall structure of the Java API packages is creative, original, and "resembles a taxonomy," Judge Alsup nonetheless concluded that it functions as "a command structure, a system or method of operation—a long hierarchy of over six thousand commands to carry out pre-assigned [sic] functions."[188] Applying copyright's limiting doctrines as the Ninth Circuit interprets them[189] and following CONTU's

[186] Google did include a small (9 lines of a 3,179–line function), "innocent," and "inconsequential" segment of code (rangeCheck) in Android and eight test files that were never introduced into Android. *See Oracle I*, 872 F. Supp. 2d at 982–83. The parties stipulated, however, that there were no damages associated with these relatively modest code portions to clear the way for appeal. *See* Final Judgment, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed June 20, 2012).

[187] *Oracle I*, 872 F. Supp. 2d at 979 (emphasis in original). *See id.* at 981 ("In order to declare a particular *functionality*, the [Java] language *demands* that the method declaration take a particular form.") (emphasis in original); *id.* at 982 (explaining that "the names of the methods and the way in which the methods are grouped" have to be the same in order to "be interoperable. Specifically, code written for one API would not run on an API organized differently, for the name structure itself dictates the precise form of command to call up any given method.").

[188] *See id.* at 999–1000.

[189] Judge Alsup placed particular emphasis on *Sega Enterprises Ltd. v. Accolade, Inc.* for its rejection of the Third Circuit's broad protection for the SSO of computer software. *See* 977 F.2d 1510, 1524–25 (9th Cir. 1992) ("The *Whelan* rule . . . has been widely—and soundly—criticized as simplistic and overbroad" (citing Comput. Assocs., Inc. v. Altai, 982 F.2d 693 (2d Cir. 1992))). Judge Alsup also emphasized the *Sega* court's recognition that "the functional requirements for compatibility with [a software platform developed by another company] are not protected by copyright. 17 U.S.C. § 102(b)." *Sega*, 977 F.2d at 1522. The Ninth Circuit expressly endorsed the Second Circuit's *Altai* approach:

> Under a test that breaks down a computer program into its component subroutines and sub-subroutines and then identifies the idea or core functional element of each, such as the test recently adopted by the

guidance that when specific computer 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*,"[190] Judge Alsup determined that Google was free to write code that accomplished the same functionality as the Java APIs at issue even if it did not achieve complete compatibility with the full Java platform.[191] In essence, Judge Alsup decided, later developers can achieve the *particular functionality* or method of operation of an API subsystem (and even groups of subsystems) so long as they write their own code and the method of writing code is not protected by a patent.

Judge Alsup's framework provided a general and concrete solution to the API copyright puzzle. Although he cautioned that his opinion was limited to the facts of the case and did not declare APIs uncopyrightable, Judge Alsup's analysis illuminated a clear pathway for software developers seeking to use APIs defined and first implemented by other software companies without running afoul of copyright law.[192] Later developers can legally use declaring code so long as they use a clean–room to implement the declarations. To many in the software industry, the ruling validated what was considered a best practice.[193] To others, it jeopardized substantial efforts and investments in developing software platforms and pioneering products, and it also threatened to undermine interoperability.[194]

> Second Circuit in *CAI*, 23 U.S.P.Q.2d at 1252–53, *many aspects of the program are not protected by copyright*. In our view, in light of the essentially utilitarian nature of computer programs, the Second Circuit's approach is an appropriate one.

Sega, 977 F.2d at 1525 (emphasis added).

[190] *Oracle I*, 872 F. Supp. 2d at 986 (emphasis added by Judge Alsup) (quoting CONTU FINAL REPORT, *supra* note 21, at 20).

[191] *Id.* at 1000.

[192] Patent protection, trade secret law, and contractual limitations could nonetheless stand in the way, but copyright protection could not bar re–implementation of functional features of computer programs. *See* J. Jonas Anderson, *Secret Inventions*, 26 BERKELEY TECH. L.J. 917, 922 (2011); *cf.* Christian Chessman, *A "Source" of Error: Computer Code, Criminal Defendants, and the Constitution*, 105 CALIF. L. REV. 179, 210 & n.224 (2017) (noting limits to trade secret protection for software given widespread commercial reliance on open–source code).

[193] *See* Wingfield & Hardy, *supra* note 2; *supra* note 69.

[194] *See* Annette Hurst, *Oracle Attorney Says Google's Court Victory Might Kill the GPL*, ARS TECHNICA (May 27, 2016, 2:35 PM), http://arstechnica.com/tech-policy/2016/05/op-ed-oracle-attorney-says-googles-court-victory-might-kill-the-gpl/; Florian Mueller, *Google's 'Fair Use' Defense Against Oracle Is an Insult to Human Intelligence: Android's Use of Java APIs Violates Copyright*, FOSS PATENTS (May 22, 2016), http://www.fosspatents.com/2016/05/googles-fair-use-defense-against-oracle.html.

2. *Federal Circuit Reversal*

Notwithstanding that Oracle did not appeal any patent issue, it filed its appeal with the U.S. Court of Appeals for the Federal Circuit. The Federal Circuit has exclusive jurisdiction over appeals from district court cases involving patent infringement allegations even though, as was the circumstance in *Oracle v. Google*, neither party challenged the district court's patent rulings. The Federal Circuit is bound by regional circuit law when reviewing questions that involve law and precedent not exclusively assigned to the Federal Circuit.[195] Thus, the Federal Circuit was required to review the copyright issues according to Ninth Circuit precedents.[196]

The "software as creative expression" theme resonated with the Federal Circuit. The court's opinion highlighted the creativity of the Java APIs.[197] The court pointed to the testimony of Joshua Bloch, the former Sun software engineer whom Google referred to as its "Java guru," who "conceded" that there can be "creativity and artistry even in a single method declaration."[198] The Federal Circuit offered its own literary metaphor, noting that "the opening of Charles Dickens' *A Tale of Two Cities* is nothing but a string of short phrases. Yet no one could contend that this portion of Dickens' work is unworthy of copyright protection because it can be broken into those shorter constituent components."[199]

The Federal Circuit reversed the district court's determination that the structure, sequence, and organization of the 37 Java APIs were not copyrightable and remanded the fair use issue for retrial with revised jury

[195] *See* Atari Games Corp. v. Nintendo of Am., Inc., 897 F.2d 1572, 1575 (Fed. Cir. 1990).

[196] Copyright issues are not within the Federal Circuit's exclusive jurisdiction. *See* 28 U.S.C. § 1295 (2012).

[197] *See Oracle II*, 750 F.3d 1339, 1352 (Fed. Cir. 2014) (explaining that the district court "acknowledged that the overall structure of Oracle's API packages is creative"); *id.* at 1356 ("The testimony at trial revealed that designing the Java API packages was a creative process and that the Sun/Oracle developers had a vast range of options for the structure and organization."); *id.* ("In its copyrightability decision, the district court specifically found that the API packages are both creative and original, and Google concedes on appeal that the originality requirements are met."); *id.* at 999 ("Yes, it is creative. Yes, it is original."); *see id.* at 1361 n.6 (noting that the Amicus Brief filed by Scott McNealy and Brian Sutphin "provide[d] a detailed example of the creative choices involved in designing a Java package"); *id.* at 1368 n.14 ("Amici McNealy and Sutphin explain that 'a quick examination of other programming environments shows that creators of other development platforms provide the same functions with wholly different creative choices.'").

[198] *Oracle II*, 750 F.3d at 1339.

[199] *Id.*

instructions. In reviewing the district court's determination that the Java API packages at issue were not copyrightable, the Federal Circuit distinguished between copyrightability of the declaring code and copyrightability of the structure, sequence, and organization of the API packages. The Federal Circuit ruled that the district court should not have considered the merger and *scènes à faire* doctrines when evaluating copyright subsistence because the Ninth Circuit treats these doctrines as affirmative defenses to infringement, not as limitations on copyrightability.[200] Hence, these doctrines were relevant only in determining what elements of the APIs should be filtered out in the infringement analysis.[201] Furthermore, the Federal Circuit held that the merger doctrine—which bars protection where an idea can only be expressed in one or a limited number of ways—properly focuses on the creative choices available to Sun when it created Java, not on the options available to Google when it copied Java APIs.[202] The Federal Circuit also held that the short phrases doctrine did not bar copyright protection for compilations of words and short phrases as reflected in declaring code.[203] Consequently, the appellate court ruled that copyright law protected the 7,000 lines of declaring code.

The Federal Circuit faulted the district court's reliance on *Lotus v. Borland*,[204] the First Circuit case holding that the Lotus 1–2–3 menu command hierarchy was an unprotectable "method of operation." The Federal Circuit distinguished *Lotus* on factual grounds, noting that the

[200] *See id.* at 1358 (citing Ets–Hokin v. Skyy Spirits, Inc., 225 F.3d 1068, 1082 (9th Cir. 2000); Satava v. Lowry, 323 F.3d 805, 810 n.3 (9th Cir. 2003)) ("The Ninth Circuit treats *scènes à faire* as a defense to infringement rather than as a barrier to copyrightability.").

[201] *See Oracle II*, 750 F.3d at 1359–62 (addressing the merger doctrine); *id*. at 1363–64 (addressing the *scènes à faire* doctrine, which Judge Alsup had rejected as a basis for holding the Java APIs to be unprotectable but that Google challenged on appeal).

[202] *See id.* at 1360–61.

[203] *See id.* at 1362–63. It should be noted that the district court's determination that the declaring code was uncopyrightable did not turn on the short–phrases doctrine. Judge Alsup recognized that the selection and arrangement of short phrases could be protectable. *See* Oracle Am., Inc. v. Google, Inc., 872 F. Supp. 2d 974, 992 (N.D. Cal. 2012) (quoting Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., Inc., 499 U.S. 340, 349 (1991), for the proposition that even thinly protected, factual compilations are protectable with respect to original "selection and arrangement"). Judge Alsup's ultimate determination turned on § 102(b) of the Copyright Act and interoperability. *See Oracle I*, 872 F. Supp. 2d at 997–1002.

[204] Lotus Dev. Corp. v. Borland Int'l., Inc., 49 F.3d 807 (1st Cir. 1995), *aff'd without opinion by equally divided court*, 516 U.S. 233 (1996).

command labels at issue there, unlike the Java API declaring code, were "not creative" and were "essential" to operating the computer system.[205] Moreover, the Federal Circuit interpreted the Ninth Circuit's cursory opinion in *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*,[206] to hold that the SSO of a computer program is eligible for copyright protection; hence it was inconsistent with *Lotus*.[207] In so doing, the Federal Circuit resurrected the flawed analysis in the Third Circuit's *Apple* and *Whelan* cases: analyzing copyrightability of computer software based on whether the high–level function(s) of the software could be implemented in multiple ways rather than viewing a particularized set of software functions as an unprotectable "method of operation."[208]

The Federal Circuit rejected the district court's invocation of interoperability as a basis for holding the SSO of the Java APIs to be uncopyrightable. Notwithstanding the language in *Sega v. Accolade* and *Sony v. Connectix*, indicating that the precise coding to achieve interoperability is not protectable under copyright law,[209] the appellate court distinguished these cases as "focused on fair use, not copyrightability."[210] The Federal Circuit held that "copyrightability is focused on the choices available to the plaintiff at the time the computer program was created," not the defendant's desire to achieve interoperability.[211] Thus, the Federal

[205] *See Oracle II*, 750 F.3d at 1365.

[206] *See* Johnson Controls, Inc. v. Phx. Control Sys., 886 F.2d 1173 (9th Cir. 1989).

[207] *See Oracle II*, 750 F.3d at 1365–66. The Federal Circuit's interpretation of *Johnson Controls* stretches its holding and overlooks important insights from later Ninth Circuit cases.

[208] *See id.* at 1366–67.

[209] *See* Sega Enters., Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992); Sony Comput. Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 603 (9th Cir. 2000) ("There is no question that the Sony BIOS contains unprotected functional elements.").

[210] *See Oracle II*, 750 F.3d at 1369 (observing that *Sega* and *Sony* never addressed whether the functional code had separable expressive elements). This assertion overlooks, however, that both courts recognized that the code that was necessary for interoperability was unprotectable and hence the copying of the entirety of the software for purposes of reverse engineering the code to determine those interoperable features constituted fair use. *See* Pamela Samuelson, *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, 31 BERKELEY TECH. L.J. 1215, 1278 & n.364 (2017) (noting congressional intent to facilitate reverse engineering for purposes of determining interoperability).

[211] *See Oracle II*, 750 F.3d at 1370 ("[A] defendant's desire 'to achieve total compatibility . . . is a commercial and competitive objective which does not enter into the . . . issue of whether particular ideas and expressions have merged . . .'" (quoting Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1253 (3d Cir. 1983))). The

Circuit explained that Google's interoperability argument comes into play only as part of a fair use defense.

The Federal Circuit leaned toward ruling in Oracle's favor on fair use, noting that "[o]n many of [Oracle's] points, Google does not debate Oracle's characterization of its conduct, nor could it on the record evidence" and that "Google knowingly and illicitly copied a creative work to further its own commercial purposes, and did so verbatim, and did so to the detriment of Oracle's market position."[212] Nonetheless, the Federal Circuit remanded the case because material facts were disputed, notably the transformativeness of the Android platform, Google's interoperability objectives, and the commercial impact of Android on Sun's/Oracle's mobile licensing activities and the potential market for a Java smartphone.[213] The Federal Circuit instructed the district court to "revisit and revise its jury instructions on fair use consistent with [the Federal Circuit's] opinion."[214]

3. The Interlocutory Certiorari Petition

Rather than seeking en banc review of the Federal Circuit's decision, Google filed a petition for a writ of certiorari with the U.S. Supreme Court.[215] Google's petition pressed the argument that the Java API declarations fall within the § 102(b) exclusion from copyright protection of methods of operation.[216] Oracle responded that the case was not appropriate for interlocutory review on substantive and prudential grounds.[217] The Supreme Court nonetheless requested the views of the Solicitor General,[218] which produced perhaps the case's most surprising filing.[219] The Solicitor General not only recommended against granting review on prudential

Federal Circuit follows the Third Circuit's dicta and not the apparent rejection of that position in *Sega* and *Sony*. *See Sega*, 977 F.2d at 1525; *Sony*, 203 F.3d at 603.

[212] *See Oracle II*, 750 F.3d at 1376.

[213] *See id*. at 1377.

[214] *See id*.

[215] *See* Petition for a Writ of Certiorari, Google, Inc. v. Oracle Am., Inc., 135 S. Ct. 2887 (2015), *denying cert. to* 750 F.3d 1339 (Fed. Cir. 2014) (No. 14-410), 2014 WL 5319724.

[216] *See id.*

[217] *See* Brief in Opposition, Google, Inc. v. Oracle Am., Inc., 135 S. Ct. 2887 (2015) (No. 14-410), 2014 WL 7205172.

[218] *See* Google, Inc. v. Oracle Am., Inc., 135 S. Ct. 2887 (2015), *denying cert. to* 750 F.3d 1339 (Fed. Cir. 2014) (No. 14–410) [hereinafter *Oracle III*].

[219] *See* Brief for the United States as Amicus Curiae, Google, Inc. v. Oracle Am., Inc., 135 S. Ct. 2887 (2015) (No. 14-410), 2015 WL 2457656.

grounds, but also sided with Oracle on substantive grounds.[220] The Supreme Court denied review.[221]

4. The 2016 Fair Use Retrial

The API copyright battle returned to Judge Alsup's court for a jury trial focused on applying "'the most troublesome [doctrine] in the whole law of copyright.'"[222] Google planned to assert again equitable–estoppel and laches defenses.[223] Oracle expanded the scope of its complaint to account for new Android versions, its expansion into new product areas (clothing, television, automobile, appliances, and media (Google Play)), and Android's dramatic market growth.[224]

Leading up to trial, the parties squabbled over the jury instructions on fair use.[225] After Judge Alsup adjusted the draft instructions based on input from the parties, one of the most momentous fair use jury trials in U.S. history commenced. Judge Alsup instructed the jury at the outset of the trial about the contours of the fair use doctrine, noting that the doctrine is an "equitable rule of reason" for which no generally accepted definition is possible.[226] He then read the statutory provision[227] and explained the four

[220] *See id.* at 11–17; *cf.* Dan Levine & Lawrence Hurley, *Google Versus Oracle Case Exposes Differences Within Obama Administration*, REUTERS (May 15, 2015, 1:08 PM), http://www.reuters.com/article/us-google-oracle-lawsuit-insight-idUSKBN0O017Z2015 0515.

[221] *Oracle III*, 135 S. Ct. at 2887.

[222] *See Oracle II*, 750 F.3d at 1372 (quoting Monge v. Maya Magazines, Inc., 688 F.3d 1164, 1170 (9th Cir. 2012) (quoting Dellar v. Samuel Goldwyn, Inc., 104 F.2d 661, 662 (2d Cir. 1939) (per curiam))).

[223] *See* Google's Trial Brief at 11–12, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 2986341 (asserting that Sun's public statements and acts approving of Android's use of Java bar enforcement of its copyrights); Order Re Willfulness and Bifurcation, Oracle Am., Inc. v. Google Inc., 131 F. Supp. 3d 946 (N.D. Cal. 2012). The equitable defenses were bifurcated and hence did not arise during the fair use trial.

[224] *See* Plaintiff Oracle's [Proposed] Supplemental Complaint, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 946 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA), 2015 WL 5305164.

[225] *See, e.g.*, Oracle Am., Inc., v. Google Inc., 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (rejecting Google's request to include "as part of a broader work" within the jury instruction defining "transformative"); Oracle's Response to the Court's Request for Critique Re Instructions on Fair Use, Oracle Am., Inc. v. Google Inc., 3:10-cv-03561-WHA (filed Apr. 14, 2016).

[226] *See* Penultimate Jury Instruction on Fair Use, Oracle Am., Inc. v. Google Inc., 3:10-cv-03561-WHA (Document 1790, filed May 3, 2016).

[227] *See* 17 U.S.C. § 107 (2012).

factors, boiling down the subtleties of the vast fair use jurisprudence into about a dozen treatise–like paragraphs.

The trial played out over eight grueling days of testimony ranging from the dramatic (embarrassing emails) to the mind–numbing (experts and fact witnesses explaining API design, open source, GNU, GPL, virtual machines, and distinctions between declaring and implementing code).[228] The jurors were treated to creative and strained analogies (filing cabinets, breakfast menus featuring hamburgers, and *Harry Potter* novels), all manner of demonstrative exhibits, and a witness list featuring some of Silicon Valley's most celebrated billionaires. Economic experts opined about transformativeness (from an economic, as opposed to a legal, perspective) and network effects. Both sides made witnesses squirm. The connection of some lines of questioning to copyright law's fair use factors was often tenuous. For example, Oracle devoted much of its trial time to exposing emails sent among Google engineers suggesting that they thought copyright law protected the Java APIs.

Given the large stakes—Oracle sought upwards of $10 billion in damages and injunctive relief—both sides employed top–notch trial teams and spared little expense. Building on the infringement ruling revived by the Federal Circuit, Oracle opened the second trial with a Johnnie Cochranesque rhyme: Google copied the heart of the Java platform to enter the mobile marketplace quickly and now pleads the "fair use excuse" to avoid the consequences.[229] Oracle argued that internal emails showed that Google took illegal "shortcuts" to create Android. Drawing on its successful Federal Circuit strategy, Oracle characterized the crafting of the Java API code as highly creative, and Google's copying of Java APIs as slavish and not transformative. Oracle characterized the Android team's decision to

[228] Media and bloggers covered the trial closely. *See generally* Joe Mullin, ARS TECHNICA, https://arstechnica.com/author/joe-mullin/ (last visited Oct. 8, 2017); Sarah Jeong, STORIFY (Twitter username @Motherboard), https://storify.com/sarahjeong (last visited Oct. 8, 2017); FOSS PATENTS, http://www.fosspatents.com/ (last visited Oct. 8, 2017) (a blog published by Florian Mueller, a self–described "intellectual property activist")*;* Dante D'Orazio et al., *Google, Oracle and Java: From a Patent Spat to a Copyright Conundrum*, VERGE, https://www.theverge.com/2012/4/13/2945536/google-oracle-java-patent-copyright-lawsuit (last visited Oct. 8, 2017) (listing forty–nine articles over six years covering the case). The author also reviewed many of the exhibits that became publicly available, such as pleadings, jury instructions, and slide decks.

[229] *See* Joe Mullin, *Google Took Our Property—And Our Opportunity, Oracle Tells Jury*, ARS TECHNICA (May 10, 2016), http://arstechnica.com/tech-policy/2016/05/oracle-tells-jury-dont-buy-googles-fair-use-excuse/.

forgo a license as underhanded—and breaking the WORA interoperability promise.

Google responded by emphasizing its hard work and large investment in building a transformative smartphone platform—bringing the functionality of robust web browsing, apps, and a host of other functionalities such as cameras and games (e.g., Angry Birds) to mobile devices.[230] It justified its use of Java based in part on Sun's encouragement of the developer community to use Java and its APIs. Google downplayed the expressive creativity of Java APIs by analogizing the API packages to the labels on a filing cabinet.[231] Google closed the trial by suggesting that transformativeness provides the sensible middle ground between stealing and free. "You don't have to choose between commercial and transformative . . . . [b]ecause the whole purpose of fair use is to promote innovation."

Following three days of deliberation, the jury found that Google had "shown by a preponderance of the evidence that its use in Android of the declaring lines of code and their structure, sequence, and organization from Java 2 Standard Edition Version 1.4 and Java 2 Standard Edition Version 5.0 constitutes a 'fair use' under the Copyright Act."[232] The verdict form did not ask the jury to make subsidiary factual findings.[233] With fair use decided in Google's favor, there was no need for a further damages phase. The jurors departed without comment, leaving the public and the appellate

[230] *See* Joe Mullin, *Google to Jury: Android Was Built With Our Engineers' Hard Work*, ARS TECHNICA (May 10, 2016, 10:56 AM), http://arstechnica.com/tech-policy/2016/05/google-to-jury-android-was-built-with-our-engineers-hard-work/.

[231] *See* Sarah Jeong, *In a $9 Billion Trial, Google's Secret Weapon Is a Filing Cabinet*, MOTHERBOARD (May 11, 2016), http://motherboard.vice.com/read/googles-lawyers-tried-to-explain-apis-to-a-jury-using-a-physical-filing-cabinet. This analogy was reminiscent of earlier API copyright cases, notably *Apple v. Microsoft* (desktop icons of the graphical user interface) and *Lotus v. Borland* (spreadsheet command labels). *See* Apple Comput., 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); Lotus Dev. Corp. v. Borland Int'l., Inc., 49 F.3d 807 (1st Cir. 1995), *aff'd without opinion by equally divided court*, 516 U.S. 233 (1996).

[232] *See* Special Verdict Form, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 946 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed May 7, 2012); Joe Mullin, *Google beats Oracle—Android makes "fair use" of Java APIs*, ARS TECHNICA (May 26, 2016), http://arstechnica.com/tech-policy/2016/05/google-wins-trial-against-oracle-as-jury-finds-android-is-fair-use/.

[233] *See* Special Verdict Form, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 946 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014) (3:10-cv-03561-WHA) (filed May 7, 2012).

court without a clear understanding of how they struck the fair use balance. Judge Alsup rejected Oracle's post–trial motions seeking judgment as a matter of law and a new trial based on alleged failure to comply with discovery responsibilities. Oracle has appealed the retrial to the Federal Circuit.

III. CRITIQUE OF THE FEDERAL CIRCUIT'S 2014 COPYRIGHTABILITY DECISION

The Federal Circuit's *Oracle v. Google* decision purports to apply Ninth Circuit jurisprudence in reviewing Judge Alsup's decision holding that the compilation of functions and the structure, sequence, and organization of the Java APIs were not copyrightable. As I explore at length in a related project,[234] the Federal Circuit misinterpreted § 102(b) of the Copyright Act, misconstrued the Ninth Circuit's software copyright jurisprudence, conflated expressive and technological "creativity," and applied an overly rigid approach to copyright law's limiting doctrines. This Section summarizes the main points.

A. MISINTERPRETATION OF THE COPYRIGHT ACT

The Federal Circuit's opinion takes a broad view of the scope of copyright protection for computer software, emphasizing the low originality threshold.[235] While recognizing the § 102(b) limitations, the court did not view those constraints as applicable to copyrightability.[236] Rather, the court saw § 102(b) as only applying at the infringement and defenses stages of analysis.

The Federal Circuit misread the clear language of the Copyright Act as well as the legislative history. Section 102(b) states that "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." A plain reading of the statute

[234] *See* Menell, *supra* note 8.

[235] *See Oracle II*, 750 F.3d 1339, 1354 (Fed. Cir. 2014) (explaining that the "'originality requirement is not particularly stringent.' . . . [Originality] means only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity.'" (quoting Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 345, 358 (1991))).

[236] *Oracle II*, at 1354 ("[T]he district court failed to distinguish between the threshold question of what is copyrightable—which presents a low bar—and the scope of conduct that constitutes infringing activity.").

indicates that these exclusions apply at the copyrightability stage of analysis.[237] They are also pertinent to infringement analysis and the fair use defense.

Google argued that the particular compilations of functions in Java API packages were uncopyrightable "method[s] of operation." The Federal Circuit rejected the proposition that § 102(b) can be invoked in this way, quoting a comment in the legislative history of the 1976 Act stating that § 102(b) "in no way enlarges or contracts the scope of copyright protection," but merely "restates . . . that the basic dichotomy between expression and idea remains unchanged."[238]

That dichotomy traces back to the Supreme Court's seminal decision in *Baker v. Selden*,[239] which held that the owner of copyright in a book disclosing a method of accounting could not bar others from using the methods disclosed in the book unless they had patent protection.[240] The Supreme Court did not inquire into whether there were other methods that achieved the same general purpose (bookkeeping). Rather, the Court categorically excluded any claim to a method of accounting even as it ruled that Selden's accounting book describing the method was copyrightable.[241] The CONTU Report concurs: "one is always free to make a machine

[237] Courts routinely apply the analogous separability analysis of the useful article doctrine at the copyrightability stage. *See* Menell, *supra* note 8.

[238] *Oracle II*, 750 F.3d at 1356 (quoting Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 356 (1991)).

[239] Baker v. Selden, 101 U.S. 99 (1879).

[240] *See id.* at 102 ("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.").

[241] The Federal Circuit attempts to fit *Baker v. Selden* into its atextual reading of § 102 by stating that, "The [Supreme] Court [in *Baker v. Selden*] indicated that, if it is necessary to use the forms Selden included in his books to make use of the accounting system, that use would not amount to copyright infringement. *See* [*Baker v. Selden*, 101 U.S. at 104] (noting that the public has the right to use the account-books and that, 'in using the art, the ruled lines and headings of accounts must necessarily be used as incident to it')." *Oracle II*, 750 F.3d at 1355. A faithful reading of *Baker v. Selden* recognizes that the Court held that the accounting method was uncopyrightable, not merely not infringed. *See* Nicholas M. Lampros, *Leveling Pains: Clone Gaming and the Changing Dynamics of an Industry*, 744 BERKELEY TECH. L.J. 743, 755 (2013); Lemley, *supra* note 46, at 5. That is the essence of the idea–expression dichotomy. *See id.*

perform *any* conceivable process (in the absence of a patent)" so long as one does not "take another's program."[242]

In accordance with this principle, Google was entitled to make a mobile device ("a machine") perform the same functions as a Java API package (a "conceivable process") with clean–roomed computer code (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.

B. MISREADING NINTH CIRCUIT JURISPRUDENCE

Beyond misconstruing § 102(b), the Federal Circuit's opinion diverges from the clear language and evolution of the Ninth Circuit's software copyright jurisprudence. Judge Alsup drew principally from the First Circuit's *Lotus* decision and the Ninth Circuit's *Sega* decision in framing his analysis. The Federal Circuit held that the *Lotus* decision is "inconsistent" with Ninth Circuit precedent[243] and that the *Sega* decision is inapt.[244] Neither of these interpretations, however, withstands scrutiny.

Although the Ninth Circuit has not had occasion to address the *Lotus* line of analysis specifically, it holds that software code that is necessary for interoperability is not copyrightable. In *Sega v. Accolade*, the Ninth Circuit stated, "the functional requirements for compatibility with the Genesis [platform are] aspects of Sega's programs that are not protected by copyright. 17 U.S.C. § 102(b)."[245] Such aspects of the Genesis video game platform are functional specifications of the computer system—a relatively simple API. The Ninth Circuit unequivocally ruled that the interface specification was not copyrightable, which parallels the *Lotus* analysis. The Ninth Circuit could not have cited the First Circuit's *Lotus* decision because that decision was not handed down until several years later.

Not only did the Federal Circuit misread the Ninth Circuit's *Sega* and *Sony* decisions, it embraced analyses that the Ninth Circuit has expressly rejected. By holding that the code for interoperability may be protectable, the Federal Circuit resurrects the Third Circuit's dicta in *Apple v. Franklin*: "courts have recognized that, once the plaintiff creates a copyrightable work, a defendant's desire 'to achieve total compatibility . . . is a

[242] *See* CONTU FINAL REPORT, *supra* note 21, at 20 (emphasis added).
[243] *Oracle II*, 750 F.3d at 1365.
[244] *Id.* at 1369.
[245] Sega Enters. v. Accolade, Inc., 977 F.2d 1510, 1522 (9th Cir. 1992).

commercial and competitive objective which does not enter into the . . . issue of whether particular ideas and expressions have merged.'"[246] To the contrary, the Ninth Circuit holds that copyright law does not stand in the way of achieving interoperability. As noted earlier,[247] the Third Circuit comment is dicta as Franklin Computer had copied the entirety of Apple's computer programs. More importantly, § 102(b), the CONTU Report, and the *Sega*/*Sony* decisions directly contradict the Third Circuit's proposition.

Moreover, the Federal Circuit endorsed and followed the Third Circuit's *Apple/Whelan* framework, holding that everything not necessary to the *general* purpose or function of a work is protectable expression: "We agree with Oracle that, under Ninth Circuit law, an original work—even one that serves a function—is entitled to copyright protection as long as the author had multiple ways to express the underlying idea."[248] The Federal Circuit credited Oracle's statement that it only claimed "its *particular* way of naming and organizing each of the 37 Java API packages" and that it "'cannot copyright the idea of programs that open an internet connection,' but 'it can copyright the precise strings of code used to do so, at least so long as "other language is available" to achieve the same function.'"[249] In an accompanying footnote, the court noted that Oracle's counsel explained at oral argument that Oracle "would never claim that anyone who uses a package-class-method manner of classifying violates our copyright. We don't own every conceivable way of organizing, we own only our specific expression—our specific way of naming each of these 362 methods, putting them into 36 classes, and 20 subclasses."[250] The Federal Circuit reasoned that as long as the same general functions could be accomplished using different code, then the first author's code for such general functions was protectable.[251]

[246] *See Oracle II*, 750 F.3d at 1357 (quoting Apple Comput. v. Franklin Comput. Corp., 714 F.2d 1240, 1253 (3d Cir. 1983)).

[247] *See supra* note 28 and accompanying text.

[248] *See Oracle II*, 750 F.3d at 1371 (quoting Apple Comput., Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983)); *see also id.* at 1366 (noting that the Third Circuit in *Apple v. Franklin* "focused 'on whether the idea is capable of various modes of expression' and indicated that, '[i]f other programs can be written or created which perform the same function as [i]n Apple's operating system program, then that program is an expression of the idea and hence copyrightable'" (quoting Apple Comput., Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1252 (3d Cir. 1983)).

[249] *See Oracle II*, 750 F.3d at 1367–68 (quoting from Oracle's Reply Brief).

[250] *Id.* at 1368 n.13.

[251] *See id.* at 1356 (setting the foundation for its analysis by observing that "the Sun/Oracle developers had a vast range of options for the structure and organization" of

While this mode of analysis comports with Ninth Circuit jurisprudence with regard to implementing code, it contradicts copyright law principles and Ninth Circuit precedent as regards declarations that are necessary to operate a particular computing system. Contrary to the Third Circuit's dicta in *Apple v. Franklin*, the Ninth Circuit's *Sega* and *Sony* decisions hold that the code necessary for interoperability is uncopyrightable.[252] Thus, a defendant's desire to achieve compatibility does enter into the issue of whether particular ideas and expressions have merged in the Ninth Circuit. It resolves the issue so long as the defendant independently writes the code to achieve the particular functions of the plaintiff's software. Secondly, the *Sega* decision unequivocally rejects the *Whelan* framework of simply asking whether there are multiple ways of programming a particular function: "[t]he *Whelan* rule . . . has been widely—and soundly—criticized as simplistic and overbroad."[253] The *Sega* court instead recognized that "the functional requirements for compatibility with [a software platform developed by another company] are not protected by copyright."[254]

the Java APIs); *id.* at 1360 ("We have recognized . . . applying Ninth Circuit law, that the 'unique arrangement of computer program expression . . . does not merge with the process so long as alternate expressions are available.'" (quoting Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 840 (Fed. Cir. 1992)); *id.* ("Because Nintendo produced expert testimony 'showing a multitude of different ways to generate a data stream which unlocks the NES console,' we concluded that Nintendo's specific choice of code did not merge with the process."); *id.* at 1360 n.5 ("It is undisputed that Microsoft and Apple developed mobile operating systems from scratch, using their own array of software packages."); *id.* at 1368 n.14 (referencing the amicus brief of former Sun executives explaining that "a quick examination of other programming environments [Apple's iOS and Microsoft's Windows Phone] shows that creators of other development platforms provide the same functions with wholly different creative choices").

[252] *See* Sega Enters. v. Accolade, Inc., 977 F.2d 1510, 1522 (9th Cir. 1992) (holding that "the functional requirements for compatibility with [a software platform developed by another company] are not protected by copyright" under 17 U.S.C. § 102(b)).

[253] *See id.* at 1525 (citing Comput. Assocs. Inc. v. Altai, 23 U.S.P.Q.2d at 1252 (2d Cir. 1992), *withdrawn and superseded*, 982 F.2d 693 (2d Cir. 1992)).

[254] *See id.* at 1522 (citing 17 U.S.C. § 102(b) (2012)).

C. CONFLATION OF EXPRESSIVE AND TECHNOLOGICAL "CREATIVITY"

The Federal Circuit embraced Oracle's argument (and that of former Sun executives)[255] that API design is a "creative," "noble craft"[256] entitled to robust protection. Oracle analogized API design to the drafting of *Harry Potter* novels.[257]

This argument, however, conflates idea and expression. APIs function as the levers and gears of digital machines. The declarations must be reproduced to replicate the particular functionality. Android programmers needed to reproduce the same package, class, and method names to allow their programs to respond to the same inputs and to produce the same outputs as Java API packages.

Protection for a particular combination of functions effectively monopolizes that technological solution. The digital revolution has taught us that once consumers and programmers become accustomed to a particular interface specification, robust intellectual property protection for APIs can have dramatic effects on competition and innovation.[258] The Supreme Court and Congress have determined that inventors must meet the patent law's higher thresholds of novelty, nonobviousness, and disclosure to garner protection for technological creativity. Furthermore, patent protection is limited to twenty years from the filing of the application, far less than copyright law's ninety–five year duration for corporate authors.

D. OVERLY RIGID APPROACH TO LIMITING DOCTRINES

The Federal Circuit erred by attempting to shoehorn analysis of API design into a framework designed for analyzing software code. As the *Lotus* court and Judge Alsup recognized, copyright law does not dictate a monolithic approach to all media. Copyright law has long relied on a

[255] *See* Corrected Brief of Scott McNealy and Brian Sutphin as Amici Curiae in Support of Reversal, Oracle Am., Inc. v. Google, Inc., 750 F.3d 1339 (Fed. Cir. 2014), *rev'g* 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA), 2013 WL 942809 (characterizing API design as a highly creative process in which programmers work from a limitless pallet of choices).

[256] *See* Opening Brief and Addendum of Plaintiff-Appellant at 12–13, 72, Oracle Am., Inc. v. Google, Inc., 750 F.3d 1339 (Fed. Cir. 2014), *rev'g* 872 F. Supp. 2d 974 (N.D. Cal. 2012) (3:10-cv-03561-WHA), 2013 WL 518611.

[257] *See id.*; *Oracle II*, 750 F.3d at 1356 (citing the district court's copyrightability decision for the proposition that "[t]he overall name tree of the Java API, of course, has creative elements").

[258] *See generally* CARL SHAPIRO & HAL R. VARIAN, INFORMATION RULES: A STRATEGIC GUIDE TO THE NETWORK ECONOMY 103–226 (1999); Menell, *supra* note 12.

common law approach for adapting the law to deal with new technologies and other dynamic considerations.[259] The idea–expression dichotomy provides flexibility in the domain of functional works. Courts need to be sensitive to technological nuance in applying § 102(b), and to take care in evolving the family of doctrines (merger, *scènes à faire*, idea–expression dichotomy, *Baker v. Selden*, and fair use) on which it is based.

Oracle v. Google is the first litigated copyright case since *Lotus* to focus specifically on copyright protection for API design.[260] Judge Alsup saw that although the Ninth Circuit had endorsed the *Altai* framework for cases involving implementing code, the *Oracle v. Google* case required an alternative framework to address API design. He recognized that *Lotus* provided pertinent analysis and that *Sega* addressed the uncopyrightability of code necessary for interoperability. His decision thoughtfully combined these elements to produce a sound framework.[261]

The scope of protection for computer software brought new issues to the fore. When *Sega* developed its lockout code for the Genesis game console, there were no constraints on the arbitrary string characters that it designated for the key. Just as bank customers can choose whatever PIN they like (within the field constraint of four numbers), Sega was free to choose an arbitrary string of letters, numbers, and symbols to lock and unlock its platform. The Ninth Circuit nevertheless determined that § 102(b) did not protect the lockout code because once it was "created" for use as lockout code, it became functional.

[259] *See* Menell, *supra* note 26, at 70.

[260] As noted above, the *Sega* case addressed this issue as part of a fair use analysis of intermediate copying of software code. *See supra* notes 47–52. This API design issue has, however, arisen in litigation contexts, but was not resolved by judicial decisions. As noted earlier, *see supra* note 96, Sun Microsystems sued Microsoft for breach of contract, copyright infringement, and trademark infringement relating to Microsoft's efforts to fragment the Java platform in the late 1990s. Express Logic sued Green Hills for alleged copyright infringement of the API for real–time operating system software. *See Express Logic Seeks Injunction Against Green Hills*, EE TIMES (June 12, 2006, 6:00 PM), http://www.eetimes.com/document.asp?doc_id=1161911. The arbitration panel interpreted Ninth Circuit law very similarly to Judge Alsup and found the declaring code (header files) at issue in that case to be uncopyrightable. *See* Patrick Mannion, *Ruling for Green Hills Clears Way for Copying of APIs*, EE TIMES (Aug. 21, 2007, 9:00 PM), http://www.eetimes.com/document.asp?doc_id=1166905 (reporting that the arbitration panel held that copyright laws do not extend to the functionality of APIs in a dispute involving real time operating systems). As a disclaimer, the author served as a consultant for Sun Microsystems in Sun's litigation against Microsoft and as an expert witness for Green Hills in the litigation brought by Express Logic. The author was compensated by the parties that retained him in these matters.)

[261] *See Oracle I*, 872 F. Supp. 2d 974, 984–97 (N.D. Cal. 2012).

The First Circuit reached a similar conclusion in the *Lotus* case. At the time that Lotus designed its menu command hierarchy for the Lotus 1–2–3 program, there were numerous options for labeling the functions and countless compilations of function names. Once programmers of macros for the Lotus spreadsheet became accustomed to those function names, however, the labels took on tremendous importance to users. To bestow copyright protection on such a system would potentially confer outsized market power over the particular method of operating a spreadsheet due to users' high switching costs—many had developed sophisticated macros for automating their accounting and other record keeping. The First Circuit recognized that this issue was best addressed at the copyrightability stage. Like Selden's accounting book, Lotus's spreadsheet program was entitled to copyright protection at the moment it was created (or in the case of Selden's book, when the applicable formalities at the time were met) but the method of operation (like Selden's accounting system) remained outside of copyright protection.

Although more sophisticated than an ATM PIN, the Genesis lockout code, or even Lotus's menu–command hierarchy, the declarations of the Java APIs similarly functioned as methods of operating particular digital machines—packages of functions. Judge Alsup's focus on § 102(b) and the *Lotus* court's framework better address the copyright issues in *Oracle v. Google* than the *Altai* framework, which was developed for analyzing copyright code.

By rigidly focusing on Ninth Circuit cases that treat the merger and *scènes à faire* doctrines as defenses to infringement rather than copyrightability doctrines,[262] the Federal Circuit missed the forest for the trees. Section 102(b) can operate as both a threshold doctrine and as part of the filtration step of infringement analysis. In fact, in the *Ets–Hokin* case, on which the Federal Circuit bases its analysis, the Ninth Circuit treats the bottle that is the object of the photograph in question as uncopyrightable under the useful–article doctrine, i.e., at a threshold copyrightability

[262] The Federal Circuit cites *Ets–Hokin v. Skyy Spirits, Inc.*, 225 F.3d 1068, 1073, 1082 (9th Cir. 2000) (involving photography), and *Satava v. Lowry*, 323 F.3d 805, 810 n.3 (9th Cir. 2003) (involving glass–encased jellyfish sculptures and holding that "[t]he Ninth Circuit treats *scènes à faire* as a defense to infringement rather than as a barrier to copyrightability").

level.[263] Copyright law, like the patent law's nonobviousness doctrine, does not fit a rigid mold.[264]

E. TREATING API DESIGN AS VARIABLE EXPRESSION RATHER THAN UNIQUE FUNCTION

The Federal Circuit erred in treating the set of 37 Java API declarations as "source code" rather than as the functional specifications for a particular computer system.[265] Such API design defines the particular data–processing capabilities of a particular computing machine and is necessary for another virtual machine to perform the same processes.

From a copyright standpoint, the critical question is whether a particular set of instructions, expressed in a particular way, is "the only and essential means of accomplishing a given task."[266] Alternatively, are these particular instructions, expressed in this particular way, the only way to effectuate "the actual processes or methods embodied in the program"?[267] As CONTU explained, "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."[268] The test is not whether there are multiple ways of writing code to perform a *general* purpose. Congress instead viewed the idea–expression dichotomy as enabling anyone to build a machine capable of performing any *particular* function, including those for which others had written computer code. Under the idea–expression dichotomy, copyright protection must not lock competitors out of a particular platform—only patent protection can. Copyright protection can only require that competitors write their own implementing code. If the only way to achieve such a "certain result"[269] includes literally copying even detailed textual–represented

[263] *See Ets–Hokin*, 225 F.3d at 1073, 1080.

[264] *Cf.* KSR Int'l Co. v. Teleflex, Inc., 550 U.S. 398 (2007) (reversing the Federal Circuit for applying too rigid a test—the teaching–suggestion–motivation requirement—for analyzing patent law's nonobviousness doctrine).

[265] *See Oracle II*, 750 F.3d 1339, 1368 (Fed. Cir. 2014) ("Given the [district] court's findings that the SSO is original and creative, and that the declaring code could have been written and organized in any number of ways and still have achieved the same functions, we conclude that Section 102(b) does not bar the packages from copyright protection just because they also perform functions.").

[266] *See* CONTU FINAL REPORT, *supra* note 21, at 20.

[267] *See* H.R. REP. No. 94-1476, at 57 (1976).

[268] *See* CONTU FINAL REPORT, *supra* note 21, at 20 (emphasis added).

[269] As added in the 1980 amendments, the Copyright Act defines a "computer program" as "set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (2012).

information, such as declarations, then copyright law does not stand in the way.

Google followed this path. It sought to achieve the particular functionalities of 37 Java API packages. After negotiations to license the Java APIs reached an impasse, Google independently wrote its own implementing code. Oracle does not dispute that Google needed to include the particular declarations to make its Android platform perform the particular functions of the 37 Java APIs. Thus, the Federal Circuit should have affirmed Judge Alsup's copyrightability ruling, and the case should have ended at that stage.

IV. THE *ORACLE V. GOOGLE* LITIGATION: FUTURE PATHWAYS

The *Oracle v. Google* fair use jury trial ranks among the most significant computer software intellectual property trials and copyright fair use trials in U.S. history. Yet, it did little to clarify intellectual property protection for computer software. Even though Google has prevailed thus far, the jury's fair use decision has little precedential significance. The jury's verdict in *Oracle v. Google* does not insulate other technology companies from the risk of copyright liability for independently implementing the code necessary to achieve particular functionality. Nor does it stand in the way of Oracle filing a new complaint alleging that new versions of Android infringe copyright protection for the structure of Java API packages.

The only secure safe harbors are to develop an independent platform or license the pre–existing APIs, each of which can have undesirable effects. Independent platforms raise compatibility and interoperability concerns, risk fragmentation of markets, and reduce positive externalities from network effects. The need to negotiate a license erects a barrier to entry and risks market exclusion and vertical monopolization.

Thus, notwithstanding six years of litigation and two trials, the *Oracle v. Google* litigation has contributed to, rather than quelled, confusion surrounding API copyright protection. Fair use is a highly unpredictable doctrine. Legal advisors will need to inform clients that there is no clear safe harbor for reimplementing APIs short of a license. Other trial teams will face the same troublesome doctrines when confronted with other sets of complex facts.

Furthermore, by resolving the fair use question by a simple jury verdict form,[270] the *Oracle v. Google* litigation sheds little light on the reasoning on which the jury based its decision. There were no formal factual findings. Therefore, the decision contributes little to our understanding of the fair use factors—transformativeness, commerciality, nature of the copyrighted work—or how they are balanced in the context of APIs. All we know is that Google's particular reimplementation for particular devices was fair use. As the motion for a new trial reveals,[271] however, new versions of the Android platform could provide the basis for a new copyright infringement action.

Such uncertainty can be especially problematic for technology companies. The design of a new platform requires planning. Network economics teaches that the viability and value of a platform depends critically upon its ability to leverage consumers' and programmers' familiarity with APIs.[272] Yet the current status of API copyright jurisprudence hinges liability for copyright infringement on fair use—"the most troublesome [doctrine] in the whole law of copyright."[273] And as the *Oracle v. Google* litigation has already illustrated, a jury verdict does not necessarily resolve a dispute. This is especially true in a case in which the cost of appeal is relatively low in comparison with the stakes involved and where the parties do not perceive advantages in a settlement.[274]

As Google completed its case in chief, Oracle filed a motion requesting that Judge Alsup render judgment as a matter law ("JMOL"). Judge Alsup

[270] *See* Notice of Final Charge to the Jury (Phase One) and Special Verdict Form, Oracle Am., Inc. v. Google, Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA).

[271] *See* Oracle's Rule 59 Motion for a New Trial, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 9045812.

[272] *See* Menell, *supra* note 8.

[273] *See Oracle II*, 750 F.3d at 1372 (quoting Monge v. Maya Magazines, Inc., 688 F.3d 1164, 1170 (9th Cir. 2012) (quoting Dellar v. Samuel Goldwyn, Inc., 104 F.2d 661, 662 (2d Cir. 1939) (per curiam))); *see also* PAUL GOLDSTEIN, GOLDSTEIN ON COPYRIGHT § 12.1 (3d ed. 2005) ("No copyright doctrine is less determinate than fair use."); David Nimmer, *"Fairest of Them All" and Other Fairy Tales of Fair Use*, 66 LAW & CONTEMP. PROBS. 263, 263 (2003).

[274] *See supra* notes 154–163 and accompanying text; Dan Levine, *Oracle Suit Versus Google at Settlement Impasse: Judge*, CHI. TRIB. (Apr. 2, 2012), http://articles.chicagotribune.com/2012-04-02/business/sns-rt-us-oracle-google-lawsuitbre8310zk-20120402_1_oracle-s-java-oracle-suit-google.

rejected Oracle's JMOL motion.[275] The court erred on Oracle's side in allowing an instruction on the propriety of the defendant's conduct,[276] notwithstanding that the Federal Circuit did not call attention to this consideration in its remand decision and the Supreme Court's decision in *Campbell v. Acuff–Rose Music, Inc.* downplays or jettisons this consideration.[277] Judge Alsup explained that, based on the evidence presented, the jury could well have determined that it was fair use to maintain the same structure of 37 Java API packages in the Android reimplemented packages so as to avoid the confusion that would ensue from scrambling the various functions: "avoiding cross-system babel promoted the progress of science and useful arts—or so our jury could reasonably have found."[278]

Judge Alsup rejected Oracle's arguments that Android's use of the Java APIs should have been deemed "entirely commercial" and nontransformative and that the Java APIs should have been considered "highly creative" because of the myriad ways in which the functions could have been implemented. With respect to the fourth fair use factor—the impact on the potential market for the Java platform—Judge Alsup ruled that the jury "could reasonably have found that use of the declaring lines of code (including their SSO) in Android caused no harm to the market for the copyrighted works, which were for desktop and laptop computers" and that

[275] *See* Order Denying Rule 50 Motions, Oracle Am., Inc. v. Google, Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 3181206.

[276] *See* Notice of Final Charge to the Jury (Phase One) and Special Verdict Form at § 27, Oracle Am., Inc. v. Google, Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA) (filed May 20, 2016).

[277] *See* Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 585 n.18 (1994) ("Even if good faith were central to fair use, 2 Live Crew's actions do not necessarily suggest that they believed their version was not fair use; the offer [to license the plaintiff's work] may simply have been made in a good-faith effort to avoid this litigation. If the use is otherwise fair, then no permission need be sought or granted."); PAUL GOLDSTEIN, 2 GOLDSTEIN ON COPYRIGHT § 12.2.2, at 12:44.5–12:45 (3d ed. 2005 & Supp. 2016).

[278] Order Denying Rule 50 Motions at 8–10, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 3181206. Judge Alsup further explained that intersystem consistency "differs from the interoperability point criticized by the Federal Circuit. The immediate point of cross-system consistency focuses on avoiding confusion in usage between the two systems, both of which are Java-based, not on one program written for one system being operable on the other, the point addressed by the Federal Circuit." *Id.* at 10 n.6.

the copying had little effect on licensing of Java ME beyond "the tailspin already predicted within Sun."[279] The court concluded its ruling by highlighting the contradiction between Oracle's pretrial instruction arguments—focusing on characterizing the fair use test as an equitable rule of reason affording juries broad discretion based on the contextual facts of the case—and its JMOL motion urging that the court override the jury's balancing of the fact–specific factors:

> In applying an "equitable rule of reason," our jury could reasonably have given weight to the fact that cross-system confusion would have resulted had Google scrambled the SSO and specifications. Java programmers and science and the useful arts were better served by a common set of command-type statements, just as all typists are better served by a common QWERTY keyboard.[280]

That decision did not, however, end even the trial court phase of the litigation. Oracle filed a new JMOL motion in early July that critiqued Judge Alsup's rejection of its first JMOL motion.[281] More significantly, Oracle filed a motion requesting a new trial based on Google's alleged failure to disclose its plan to install Android Marshmallow on desktop and laptop computers.[282] In its reply to Google's opposition,[283] Oracle contended that the withheld evidence "directly refutes Google's argument to the jury that 'Android is not a substitute [because] Java SE is on personal computers; Android is on smartphones.'"[284] Judge Alsup rejected these motions but left open the option for Oracle to file a new copyright infringement complaint

[279] *See id.* at 17.

[280] *See id.* at 18.

[281] *See* Oracle's Rule 50(b) Motion for Judgment as a Matter of Law, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 9045685.

[282] *See* Oracle's Rule 59 Motion for a New Trial, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 9045812.

[283] *See* Google Inc.'s Opposition to Oracle's Rule 59 Motion for a New Trial, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 9045809.

[284] *See* Oracle's Reply in Support of its Rule 59 Motion for a New Trial at 1, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 9045811.

based on Google's implementations of Android in devices other than smartphones and tablets.[285] Oracle has appealed the fair use decision.[286]

Oracle has reason for optimism about a Federal Circuit appeal.[287] Under the Federal Circuit's Internal Operating Procedures, the same panel that reversed Judge Alsup's copyrightability ruling and set forth guiding principles for the fair use trial will likely hear the appeal of the fair use trial.[288] Oracle has preserved various objections to Judge Alsup's jury instructions. Oracle can also pursue the district court's denial of its new trial motion. Should Oracle prevail, it will have the opportunity to learn from what will have become an expensive mock trial. It can potentially improve some of its themes and better prepare its witnesses. Moreover, Google will be prevented from asserting one of its key arguments—that Android is not a substitute for Java SE on personal computers. Alternatively, Google might decide to redesign its Chrome integration with Android to work around the 37 Java APIs. But even if Google does so, its fair use argument may be weakened, especially if it integrates its mobile and desktop platforms. The appellate panel has already indicated that there was much force to Oracle's position and that many of the facts relevant to fair use were not in dispute.[289]

[285] *See* Order Denying Renewed Motion for Judgment as a Matter of Law and Motion for a New Trial at 5, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 5393938.

[286] *See* Joe Mullin, *It's Official: Oracle Will Appeal Its "Fair Use" Loss Against Google*, ARS TECHNICA (Oct. 27, 2016), http://arstechnica.com/tech-policy/2016/10/its-official-oracle-will-appeal-its-fair-use-loss-against-google/.

[287] *See* Florian Mueller, Oracle v. Google: *Jury Finds in Favor of "Fair Use," as No Reasonable, Properly-Instructed Jury Could Have*, FOSS PATENTS (May 26, 2016), http://www.fosspatents.com/2016/05/oracle-v-google-jury-finds-in-favor-of.html (contending that Judge Alsup's instructions set the fair use bar far too low). *But see* Jonathan Band, *Sanity Prevails Again, Part II: The District Court Leaves the* Oracle v. Google *Fair Use Verdict in Place*, DISRUPTIVE COMPETITION PROJECT (June 10, 2016), http://www.project-disco.org/intellectual-property/061016-sanity-prevails-again-part-ii-the-district-court-leaves-the-oracle-v-google-fair-use-verdict-in-place (contending that "given how the district court meticulously found evidence in the record supporting the reasonableness of the jury's fair use finding, it is hard to imagine that the Federal Circuit will reverse it").

[288] *See* U.S. CT. OF APPEALS FOR THE FED. CIRCUIT, INTERNAL OPERATING PROCEDURES 11–12 (2008), http://www.cafc.uscourts.gov/sites/default/files/IOPs122006.pdf ("When an appeal is docketed in a case that was previously remanded by this court . . . the clerk's office attempts to assign the appeal to the previous panel, to a panel including at least two members of the previous panel (if one of those members was the authoring judge), or to a panel that contains the authoring judge, if such a panel is otherwise constituted and available on a subsequent argument calendar.").

[289] *See Oracle II*, 750 F.3d at 1376.

Google also has reason for optimism. First, it won the jury trial after Judge Alsup modified the jury instructions in light of the parties' concerns. Second, even if Google again lost at the Federal Circuit, it could petition the Supreme Court to review the Federal Circuit's API copyrightability ruling,[290] which could expand Android's reach and remove the cloud of future infringement lawsuits.

Assuming that the parties cannot reach a settlement, the Federal Circuit will review the fair use trial and post–trial rulings. Should Google prevail, Oracle would likely take a shot at Supreme Court review. Google would have the option of reasserting the copyrightability issue. Alternatively, the Federal Circuit could remand for another fair use trial or resolve the ultimate fair use question in Oracle's favor, thereby setting up a Google writ of certiorari petition raising both API–copyrightability and fair–use questions. Thus, even in the most optimistic scenario, the case will drone on for several more years.

V. DEBUGGING APPELLATE INTELLECTUAL PROPERTY JURISDICTION

The unusual jurisdictional posture of the *Oracle v. Google* case highlights an overlooked defect of appellate intellectual property jurisdiction. When Congress established the Court of Appeals for the Federal Circuit in 1982, it sought to address confusion in patent jurisprudence and the forum shopping that it generated.[291] Legislators did not, however, provide a procedure for reviewing Federal Circuit interpretations of regional circuit law short of Supreme Court review. Forum shopping motivated by conflicting regional circuit patent jurisprudence dominated the policy discussion. By contrast, computer software litigation was in its infancy, and the patentability of computer software was in flux.[292] Thus, it is not surprising that Congress did not put

[290] *See* Google's Trial Brief at 8 n.12, Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *remanded to* 118 U.S.P.Q.2d (BNA) 1561 (N.D. Cal. 2016) (3:10-cv-03561-WHA), 2016 WL 2986341 ("Google does not waive and hereby expressly preserves its position that the SSO/declarations are not protected by copyright law. *See, e.g.*, Bikram's Yoga Coll. of India, L.P. v. Evolation Yoga, LLC, 803 F.3d 1032 (9th Cir. 2015).").

[291] *See infra* notes 294–304 and accompanying text.

[292] *Compare* Diamond v. Diehr, 450 U.S. 175 (1981) (viewing software claims as a whole as eligible for patent protection so long as there is post–solution activity), *with* Parker v. Flook, 437 U.S. 584 (1978) (holding that a claim to a computer program is

in place the types of checks and balances that might be needed to avoid or limit jurisprudential confusion in non–patent aspects of the Federal Circuit's jurisdiction.

With the emergence of both software patenting and copyright protection for computer software, it was only a matter of time before Federal Circuit and regional circuit copyright jurisprudence would intersect. The *Oracle v. Google* case illustrates the "forking"[293] of Ninth Circuit copyright jurisprudence. Whereas Judge Alsup placed principal reliance on the Ninth Circuit's *Sega* decision, which expressly rejected the *Whelan* framework, the Federal Circuit emphasized the *Nintendo v. Atari Games* decision,[294] a prior Federal Circuit decision applying Ninth Circuit law. That decision predates *Sega* and builds on the inchoate foundation of the Ninth Circuit's *Johnson Controls* decision.

The *Oracle v. Google* litigation reveals the jurisprudential confusion that can arise from surrogate interpretation of judicial decisions. En banc review provides a mechanism for addressing intra–circuit splits. The only en banc process available for Google, however, would have been at the Federal Circuit. It is understandable why Google chose to pursue a writ of certiorari at the Supreme Court rather than en banc review. It is unlikely that the Federal Circuit would have seen review of a unanimous panel decision interpreting regional circuit law as justifying the significant organizational resources of en banc review. Furthermore, such review could have jeopardized collegiality among Federal Circuit jurists on questions that are outside of the Federal Circuit's principal jurisprudence.

The Federal Circuit's expansive view of API copyrightability in conjunction with the jurisdictional misalignment has been costly for Google and the larger software industry. Google has now endured a second costly trial and has had to pursue its business and technology strategy under a cloud of confusion about the copyrightability of the functions and labels

ineligible for patent protection unless the claim entails inventive elements beyond any algorithms).

[293] Forking of software code refers to creating an independent branch of a computer program. *See Fork (Software Development)*, WIKIPEDIA, https://en.wikipedia.org/wiki/Fork_(software_development) (last visited Oct. 8, 2017); Marcel T. Rosner & Andrew Kang, *Understanding and Regulating Twenty-First Century Payment Systems: The Ripple Case Study*, 114 MICH. L. REV. 649, 663 n.113 (2016). This split from the original program typically "spawns competing projects that cannot later exchange code, splitting the potential developer community." *See* Eric S. Raymond, *Promiscuous Theory, Puritan Practice*, *in* HOMESTEADING THE NOOSPHERE (2000), http://www.catb.org/~esr/writings/cathedral-bazaar/homesteading/ar01s03.html.

[294] Atari Games Corp. v. Nintendo of Am., Inc., 897 F.2d 1572 (Fed. Cir. 1990).

within API packages. The greater software industry has endured continued uncertainty about the state of a critical aspect of copyright law.

There are several approaches to fix this bug in the appellate jurisdictional system. There is no justification for routing appeals of nonpatent issues governed by regional circuit law to the Federal Circuit when patent issues are not appealed. But were Congress to amend the Federal Circuit's jurisdiction so that such appeals would go to the regional circuit, patent owners could easily circumvent that rule by appealing patent issues they might otherwise drop solely to get the Federal Circuit to review nonpatent issues. More significantly, there are many patent cases with nonpatent issues that merit appeal on both patent and non–patent grounds. Hence, the allocation of appellate jurisdiction over cases raising patent and non–patent issues will arise.

Section A traces the legislative intent underlying the Federal Circuit's subject matter jurisdiction. Section B develops a framework for assessing appellate intellectual property jurisdiction. Section C applies that framework to assess appellate intellectual property jurisdictional regimes.

A. THE FEDERAL CIRCUIT'S SUBJECT MATTER JURISDICTION

The establishment of the Federal Circuit grew out of general concern about the federal judiciary's ability to keep pace with the demands of a growing nation, global economy, and ever–expanding and increasingly complex set of laws. Federal dockets had grown significantly in the 1960s and there was widespread concern about the strain on all levels of the federal judiciary.[295]

In 1972, Congress established the Commission on Revision of the Federal Court Appellate System, Structure and Internal Procedure to study the functioning of the appellate courts and make reform recommendations.[296] The Commission proposed, among other measures,

[295] *See* PAUL D. CARRINGTON, AM. BAR FOUND., ACCOMMODATING THE WORKLOAD OF THE UNITED STATES COURTS OF APPEALS (1968); HENRY J. FRIENDLY, FEDERAL JURISDICTION: A GENERAL VIEW 31–47 (1973); COMM. ON RULES OF PRACTICE AND PROCEDURE, JUDICIAL CONFERENCE OF THE U. S., REPORT OF THE PROCEEDINGS OF THE JUDICIAL CONFERENCE OF THE UNITED STATES 38–39 (1971); Griffin B. Bell, *Toward a More Efficient Federal Appeals System*, 54 JUDICATURE 237, 237–38 (1971); Paul D. Carrington, *Crowded Dockets and the Courts of Appeals: The Threat to the Function of Review and the National Law*, 82 HARV. L. REV. 542 (1969).

[296] *See* Roman L. Hruska, *The Commission on Revision of the Federal Court Appellate System: A Legislative History*, 1974 ARIZ. ST. L.J. 579 (1974).

the establishment of a National Court of Appeals.[297] The Commission also called attention to the problem of forum shopping in the patent field[298] but did not recommend creating a specialized court for patent appeals.[299] The Commission believed that its proposed National Court of Appeals would better address the patent forum shopping concerns. The Commission's grand appellate reform proposal, however, failed to gain passage in Congress.

Several years later, growing concerns about economic stagnation led President Carter's Domestic Policy Review on Industrial Innovation to pursue a specialized patent appellate court as a means of spurring research and development.[300] Advocates for a specialized patent appellate court believed that jurisprudential divisions among the regional courts of appeal undermined investment and innovative activity.[301] Many jurists, legislators, and key bar associations resisted the creation of a specialized patent tribunal, largely on the grounds that general jurists and regional courts best serve the administration of justice.[302] Supporters of consolidating patent

[297] *See* HRUSKA COMMISSION REPORT, *supra* note 5.

[298] *See id.* at 220–21 (quoting Judge Henry Friendly describing "mad and undignified races between a patentee who wishes to sue for infringement in one circuit believed to be benign toward patents, and a user who wants to obtain a declaration of invalidity or non-infringement in one believed to be hostile to them") (citing FRIENDLY, *supra* note 295).

[299] *See id.* at 234–36.

[300] *See* H.R. REP. No. 96-1307 (1980) (diagnosing the causes of economic stagnation as the "failure of American industry to keep pace with the increased productivity of foreign competitors"); Griffin B. Bell & Terence B. Adamson, *Daniel J. Meador—Visionary*, 80 VA. L. REV. 1209, 1212–13 (1994) (describing Daniel Meador's efforts as head of Office for Improvements in the Administration of Justice to establish the Federal Circuit); Helen W. Nies, *Special Session of the United States Court of Appeals for the Federal Circuit Commemorating Its First Ten Years*, 2 FED. CIR. B.J. 267, 270 (1992) ("Professor Meador was the first to conceive the idea of the Federal Circuit. As Assistant Attorney General from 1977 to 1979, he headed the Office for Improvements in the Administration of Justice which shepherded the legislation to create this court."); *see also generally* Elizabeth I. Rogers, *The Phoenix Precedents: The Unexpected Rebirth of Regional Circuit Jurisdiction over Patent Appeals and the Need for a Considered Congressional Response*, 16 HARV. J.L. & TECH. 411, 421–30 (2003).

[301] *See* H.R. REP. No. 96-1307 (1980) (explaining that a single court for patent appeals "will do a great deal to improve investors' confidence in patented technology").

[302] *See* George C. Beighley Jr., *The Court of Appeals for the Federal Circuit: Has it Fulfilled Congressional Expectations?*, FORDHAM INTELL. PROP. MEDIA & ENT. L.J. 670, 689–90, 693–97 (2011); *see also* S. REP. 97-275, at 40–41 (1981) (providing the additional views of Senator Max Baucus), *as reprinted in* 1982 U.S.C.C.A.N. 11, 50–51 ("[T]he American Bar Association and the American College of Trial Lawyers have actively opposed that portion of S. 1700 that would remove patent appeals jurisdiction from the eleven federal circuit courts of appeals. They share my concern that creating such a

appeals in a single tribunal countered that the proposed appellate tribunal, which merged appellate responsibilities for claims against the government, trade matters, and several other areas with appeals of patent cases, belied the "specialized court" label. The proposed court would have a range of responsibilities and include generalist judges.[303]

The counterargument carried the day. Congress passed the Federal Courts Improvement Act of 1981,[304] establishing a new Article III appellate tribunal: the U.S. Court of Appeals for the Federal Circuit. Nonetheless, legislators circumscribed the Federal Circuit's exclusive jurisdiction to preserve regional circuit court primacy in nonpatent areas of law.

Congress voiced concern about the Federal Circuit expanding its exclusive patent jurisdiction to other areas, such as antitrust law. The Senate Judiciary Committee noted the risk and specifically warned against the Federal Circuit's exclusive jurisdiction over patent claims being manipulated or extended. The Senate Report explained that the establishment of the Federal Circuit:

> is intended to alleviate the serious problems of forum shopping among the regional courts of appeals on patent claims by investing exclusive jurisdiction in one court of appeals. *It is not intended to create forum shopping opportunities between the Federal Circuit and the regional courts of appeals on other claims*.[305]

The Committee noted that:

> If, for example, a patent claim is manipulatively joined to an antitrust action, but severed or dismissed before final decision of the antitrust claim, jurisdiction over the appeal of the antitrust

specialty court is not in the best interest of the legal system."); Dennis Crouch, *An Open Letter from Judge Rader*, PATENTLY-O (June 30, 2014), http://patentlyo.com/patent/2014/06/letter-judge-rader.html (expressing regret in a farewell letter to his colleagues on the Federal Circuit that as a Senate Judiciary Committee staffer in the early 1980s, he "allowed judges from the Ninth Circuit to dissuade [him] from offering an amendment to include copyright and trademark cases within the jurisdiction of the Federal Circuit.").

[303] *See* Daniel J. Meador, *Retrospective on the Federal Circuit: The First 20 Years—A Historical View*, 11 FED. CIR. B.J. 557, 558 (2001); Daniel J. Meador, *Origin of the Federal Circuit: A Personal Account*, 41 AM. U. L. REV. 581 (1992).

[304] Federal Courts Improvement Act, Pub. L. No. 97-164, 96 Stat. 25 (1982).

[305] *See* S. REP. 97-275 (1981), *as reprinted in* 1982 U.S.C.C.A.N. 11; *see also* H.R. REP. No. 96-1307, at 23 (1980) ("[J]urisdiction of an appeal in a case involving a claim arising under any Act of Congress relating to copy rights or trademarks . . . will continue to go to the regional appellate courts, pursuant to section 1294 of title 28").

> claim should not be changed by this Act but should rest with the regional court of appeals.[306]

Senator Leahy specifically warned that "[i]n nearly all . . . litigation [other than patent cases], science and technology, when relevant, are related to other human or social issues, and only a generalist court should ever hear such matters."[307]

Congress did not, however, clearly foresee the potential for jurisprudential confusion and forum shopping that could arise from the Federal Circuit's interpretation of regional circuit law. The potential for cases raising both patent and copyright questions would have seemed remote at the time that the Federal Circuit was crafted. Software litigation was in its infancy at the time, with substantial questions about the patent eligibility of computer software.[308] Even as software patenting expanded in the mid to late–1990s, the decline of software copyright litigation meant that complaints asserting both patent and copyright causes of action were rare.

The Federal Circuit's 2014 *Oracle v. Google* decision validates legislators' fears of overreach. The Federal Circuit's questionable interpretation of Ninth Circuit copyright law now motivates software intellectual property owners to bundle patent and copyright claims in order to take advantage of the Federal Circuit's expansive interpretation of software copyright protection. It is no coincidence that Cisco filed its complaint alleging software patent and copyright causes of action against Arista Networks after the Federal Circuit's 2014 *Oracle v. Google* decision.[309] The following Sections explore how the courts and Congress can ensure fidelity to regional circuit copyright law and prevent appellate forum shopping.

B. ANALYTICAL FRAMEWORK FOR ASSESSING APPELLATE INTELLECTUAL PROPERTY JURISDICTION

If we were assessing the design of intellectual property jurisdiction on a clean slate, the role for specialization and expertise would come

[306] *See* S. REP. 97-275 (1981), *as reprinted in* 1982 U.S.C.C.A.N. 11.

[307] *See* S. REP. 97-275, at Appendix B (1981), *as reprinted in* 1982 U.S.C.C.A.N. 11 (providing the additional views of Senator Patrick J. Leahy).

[308] *See supra* note 292.

[309] *See* Hardy, *supra* note 9; Graham, *supra* note 11.

centrally into play.[310] This Article, however, operates within the legislative landscape underlying the federal appellate system. There are four principal considerations within that constrained universe that guide the analysis of appellate jurisdiction of cases presenting both patent and copyright causes of action: jurisprudential integrity, federalism, specialization bias, and administrative efficiency.

1. *Jurisprudential Integrity*

The primary goal of appellate review is to ensure correct interpretation and application of the law. The traditional hierarchical nesting of district courts within regional circuits solves this problem through direct review of decisions by the regional circuit in which the district court sits. Intra–circuit splits can be addressed through en banc review. The Supreme Court provides a final judicial check and typically only intervenes to resolve inter–circuit splits.

Due to the divided appellate authority for patent and nonpatent issues, the federal judiciary comprises an overlap of appellate authority. By allocating exclusive jurisdiction to the Federal Circuit for patent appeals, the Federal Courts Improvement Act of 1981 created a form of surrogate appellate review of nonpatent issues. The Federal Circuit must interpret and apply the law of regional circuits in reviewing nonpatent issues. Such interpretive capacity is not uncommon within hierarchical judicial systems. For example, federal courts routinely interpret state and foreign law, and state courts sometimes interpret the law of other states.[311] Nonetheless, such surrogate review creates the potential for the emergence of potentially conflicting bodies of regional circuit law. If there is no effective means for checking the interpretive divergence, the integrity of regional jurisprudence is compromised.

While the interpretation and application of the law of another jurisdiction operates smoothly when the regional circuit law is settled, the task becomes difficult when the regional circuit law is inchoate, ambiguous,

[310] *See, e.g.*, Laura G. Pedraza-Fariña, *The Federal Circuit: An Expert Community Approach*, 30 BERKELEY TECH. L.J. 89 (2015); Harold C. Wegner, *Federal Circuit Exclusive Appellate Patent Jurisdiction: A Response to Chief Judge Wood*, 13 CHI.-KENT J. INTELL. PROP. 394 (2014); Diane P. Wood, *Is It Time to Abolish the Federal Circuit's Exclusive Jurisdiction in Patent Cases?*, 13 CHI.-KENT J. INTELL. PROP. 1 (2013); Paul R. Gugliuzza, *The Federal Circuit as a Federal Court*, 54 WM. & MARY L. REV. 1791 (2013); Craig A. Nard & John F. Duffy, *Rethinking Patent Law's Uniformity Principle*, 101 NW. U. L. REV. 1619 (2007).

[311] *See* Donald H. Zeigler, *Gazing into the Crystal Ball: Reflections on the Standards State Judges Should Use to Ascertain Federal Law*, 40 WM. & MARY L. REV. 1143 (1999).

or evolving. In such circumstances, Federal Circuit review of nonpatent issues creates a risk of divergent interpretation of regional circuit law.

2. *Federalism*

The division of responsibility among regional circuits reflects political struggles and compromises dating to the nation's founding. Deep divisions among the nation's founders hampered the establishment of a coherent intermediate appellate system for more than a century. At the nation's founding and continuing to some extent to this day, Federalists and Anti–Federalists divided over the extent of federal power. Federalists advocated a substantial national government and a strong lower federal judiciary. Anti–Federalists sought to weaken federal power, including judicial authority.[312] They advocated passage of a Bill of Rights to protect citizens against the tyranny of national government and preferred to leave judicial power within state government. The clash of perspectives played out in the First Congress in 1789, resulting in a grand compromise that produced the Bill of Rights and a limited system of lower federal courts tied to state boundaries.[313]

The 1789 Judiciary Act established three judicial levels—district courts, circuit courts, and, as set forth in the Constitution, the Supreme Court. The district courts and Supreme Court corresponded roughly to their modern forms. Each state had a single district court. District court jurisdiction, however, was limited and far narrower than the Constitution authorized. Congress authorized federal district courts to adjudicate admiralty, diversity of citizenship, federal criminal, and U.S. plaintiff cases. The original U.S. Supreme Court had a Chief Justice and five associate justices.

The early circuit courts, however, were very different from their contemporary counterparts. The jurisdiction of the circuit courts was limited to cases involving diversity of citizenship, major federal crimes, cases brought by the U.S. government, and larger civil and admiralty cases. The three circuit courts (one for northeastern districts, one for central Atlantic

[312] *See generally* RUSSELL R. WHEELER & CYNTHIA HARRISON, CREATING THE FEDERAL JUDICIAL SYSTEM (3d ed. 2005) (tracing the history of the federal judiciary).

[313] Reflecting the complexity and dynamism of the issues and the times, James Madison, an early Federalist and advocate for ratification of the United States Constitution as the foundation for a strong national power, *see* THE FEDERALIST NO. 10 (James Madison), broke with Alexander Hamilton and the Federalist Party in 1891 and organized the Democratic–Republican Party with Thomas Jefferson. He played a central role in drafting and ratifying the Bill of Rights, a cornerstone of the Anti–Federalists' effort to weaken national power.

states, and one for southern states) sat twice each year in one or two specified cities of each district. The circuit panel comprised two Supreme Court justices assigned to that circuit (hence the phrase "riding circuit") and the district judge in that district; initially, there was only one district judge authorized for each district (i.e., for each state).

As the United States' geographical reach and national economy developed, the jurisdiction and size of the federal judiciary grew.[314] The need for more effective judicial administration increased further as federal law expanded. For much of the first century of the United States judiciary, the circuit courts operated principally through Supreme Court justices "circuit riding" among the districts and hearing appeals in conjunction with district judges.[315]

In his first State of the Union message to Congress in 1861, President Lincoln declared, "the country has outgrown our present judicial system."[316] He noted that the eight recently admitted states had never had "circuit courts attended by supreme judges" and that adding enough justices to the Supreme Court to accommodate all the circuit courts that were needed would make the Supreme Court "altogether too numerous for a judicial body of any sort." Lincoln proposed fixing the Supreme Court at a "convenient number," irrespective of the number of circuits and dividing the country "into circuits of convenient size." The circuit courts could be served by either Supreme Court justices or judges appointed specifically for the circuit courts.

As the backlog of appeals grew, Congress eventually established nine circuit judgeships in 1869, far below the number needed to handle the mushrooming appellate backlog. The Judiciary Act of 1875 expanded federal jurisdiction to include federal questions and cases alleging more than $500 in controversy. Growing dockets and budgetary pressures

[314] *See* WHEELER & HARRISON, *supra* note 312, at 9. Congress doubled the number of circuit courts in 1802, with one Supreme Court justice assigned to each circuit. Act of Apr. 29, 1802, 2 Stat. 118 (1802). As the number of states and territories expanded, necessitating additional district and circuit courts, Congress expanded the number of Supreme Court justices accordingly. In 1863, Congress created a tenth seat on the Supreme Court, Act of Mar. 3, 1863, 12 Stat. 794 (1863) ("To provide Circuit Courts for the Districts of California and Oregon . . . ."), although the full court rarely convened due to illness and vacancies. *See* Carl B. Swisher, *The Taney Period, 1836–64*, *in* 5 THE HISTORY OF THE SUPREME COURT OF THE UNITED STATES 839 (1974). Six years later Congress set the number of justices at nine, *see* Judiciary Act of 1869, 16 Stat. 44, where it has remained.

[315] *See* WHEELER & HARRISON, *supra* note 312, at 7–19.

[316] *See* Abraham Lincoln, *Message to Congress of Dec. 3, 1861*, *in* 5 THE COLLECTED WORKS OF ABRAHAM LINCOLN 41 (R. Basler ed., 1953).

strained the federal judiciary. Much of the burden fell to the 65–odd district judges, who were hearing close to 90 percent of the appeals by the 1880s, in addition to their large and growing trial court responsibilities.[317] Furthermore, the Supreme Court was obliged to hear almost all cases in which litigants sought high court review, resulting in a massive logjam at the top of the federal judiciary pyramid.[318]

Thus, a century after its establishment, the federal judiciary was in crisis.[319] Supreme Court justices had long abandoned riding circuit. The ranks of intermediate circuit judgeships were inadequate to handle the rising appellate caseload, adding substantial additional burden to an overextended district judge corps. Moreover, broad access to the Supreme Court impeded its capacity to review cases with alacrity.

Dissatisfaction with the operation of the federal judiciary ultimately led Congress to pass the Circuit Court of Appeals Act of 1891,[320] commonly referred to as the Evarts Act, establishing the modern circuit court system. Senator Evarts orchestrated a compromise that increased the role of the national courts while preserving state and regional influence.[321] Furthermore, the legislation substantially shifted the Supreme Court's workload to nine separate regional circuit courts of appeals and authorized the appointment of 19 circuit court judges, three for the Second Circuit and two for each of the others.

The relatively small scale of appellate courts—initially 19 circuit judges among the nine regional circuits—functioned smoothly during the first several decades, leaving the Supreme Court to focus on inter–circuit splits. As Congress expanded the size of the appellate judiciary to address growing caseloads, the problem of intra–circuit splits emerged. The Supreme Court addressed these issues by authorizing en banc review of intra–circuit

[317] *See* FELIX FRANKFURTER & JAMES LANDIS, THE BUSINESS OF THE SUPREME COURT 60, 79 (1928) (reporting that the number of cases pending in the federal courts rose 86%—from 29,000 to 54,000—between 1873 and 1890).

[318] By 1890, the Supreme Court had 1,816 cases on its docket, including 623 cases filed that year. *See id.* at 101–02.

[319] *See* PAUL M. BATOR ET AL., HART & WECHSLER'S THE FEDERAL COURTS AND THE FEDERAL SYSTEM 37 (3d ed. 1988) (referring to the post–Civil War period as "the nadir of federal judicial administration").

[320] *See* Evarts Act, Act of Mar. 3, 1891, 26 Stat. 826 (1891); WHEELER & HARRISON, *supra* note 312, at 16–18; *See* ERWIN C. SURRENCY, HISTORY OF THE FEDERAL COURTS 49 (1987).

[321] *See* WHEELER & HARRISON, *supra* note 312, at 16–18.

conflicts.[322] Congress codified the Supreme Court's *Textile Mills* decision in passing the Judicial Code of 1948.[323]

Congress's decision to centralize and consolidate appellate patent review consciously diverged from the federalist structure of the appellate courts, for the purpose of eliminating the interpretive confusion and forum shopping that had emerged in patent cases. Yet, federalism concerns were voiced during consideration of the Federal Courts Improvement Act[324] and Congress retained the federalist judicial structure for non–patent issues. The failure to provide a mechanism to ensure fidelity to regional circuit law, however, creates the potential for the confusion manifested in the *Oracle v. Google* litigation.

[322] *See* Textile Mills Sec. Corp. v. Comm'r, 314 U.S. 326 (1941) (interpreting the Judicial Code to permit en banc review). Congress supplanted the Evarts Act by passing the Judicial Code in 1911. *See* Act of Mar. 3, 1911, ch. 231, 36 Stat. 1087.

[323] Section 46(c) of the Judicial Code of 1948 provided that circuit courts could convene en banc panels upon a majority vote of active judges in the circuit. "A court in banc shall consist of all active circuit judges of the circuit." Act of June 25, 1948, ch. 646, 62 Stat. 871 (1948). Congress left the specific procedures and standards for doing so, however, to the circuit courts. *See* Pub. L. No. 88-176, 77 Stat. 331 (1963).

[324] *See, e.g.*, S. REP. 97-275, at 40–41 (1981) (quoting HRUSKA COMMISSION REPORT, *supra* note 5):

> Giving a national court exclusive jurisdiction over appeals in a category of cases now heard by the circuit courts would tend to dilute or eliminate regional influence in the decision of those cases. Our nation is not yet so homogenous that the diversity of our people cannot be reflected to some advantage in the decisions of the regional courts. Excluding these courts from consideration of particular categories of cases would also contract the breadth of experience and knowledge which the circuit judges would bring to bear on other cases; the advantages of decision making by generalist judges diminish as the judge's exposure to varied areas of the law is lessened.);

Id.; *see also id.* (additional Views of Senator Max Baucus) ("Many of us in the Congress have been greatly disturbed by the growing trend toward centralizing decision making in Washington, D.C. Many of us have supported venue reform to ensure that cases are litigated in States, where the problems arise, rather than in the District of Columbia. Similarly, I believe that we must avoid centralized specialty courts."); S. REP. 97-275, at Appendix B (1981), *as reprinted in* 1982 U.S.C.C.A.N. 11 (additional Views of Senator Patrick J. Leahy on S. 1700) (advocating creation of the Federal Circuit, but agreeing with "the concerns expressed about the precedent of establishing specialty courts, which in general would be very detrimental to our tradition of diversity and independence on the bench").

3. *Specialization Bias*

Political scientists, legal scholars, and jurists have long worried that specialized courts are more prone to political influences[325] and tunnel vision[326] than courts of general jurisdiction.[327] The legislative record shows that corporate interests played a large role in creating the Federal Circuit.[328] In a study of the Court of Customs and Patent Appeals (CCPA), one of the courts merged into the Federal Circuit that handled appeals of patent examination, Professor Lawrence Baum found that:

> [t]he patent specialists on the court, appointed through the efforts of the patent bar, have led the CCPA to adopt a line of policy significantly different from the patent policies that prevail in most of the federal judiciary. The CCPA's specialization ultimately has

[325] *See* H.R. REP. No. 97-312, at 31 (1981) (noting that "[s]everal witnesses . . . expressed fears that the Court of Appeals for the Federal Circuit would be unduly specialized or would soon be captured by specialized interests"); *see also* Lawrence Baum, *Judicial Specialization, Litigant Influence, and Substantive Policy: The Court of Customs and Patent Appeals*, 11 LAW & SOC'Y REV. 823 (1977) (arguing that court specialization enhances the likelihood of litigant interest groups affecting substantive policy).

[326] *See* Rochelle Cooper Dreyfuss, *The Federal Circuit: A Case Study in Specialized Courts*, 64 N.Y.U. L. REV. 1, 3 (1989); Alan B. Parker, *Examining Distinctive Jurisprudence in the Federal Circuit: Consequences of a Specialized Court*, 3 AKRON INTELL. PROP. J. 269, 287–89 (2009) (discussing concerns of doctrinal, intellectual, and judicial isolation); Simon Rifkind, *A Special Court for Patent Litigation? The Danger of a Specialized Judiciary*, 37 A.B.A. J. 425, 425–26 (1951) (expressing concern that the specialization and "seclusiveness" of patent law "immunizes it against the refreshment of new ideas, suggestions, adjustments and compromises which constitute the very tissue of any living system of law").

[327] *See generally* WILLIAM M. LANDES & RICHARD A. POSNER, THE POLITICAL ECONOMY OF INTELLECTUAL PROPERTY LAW 26–27 (2004) (arguing that "a specialized court is more likely to have a 'mission' orientation than a generalist court"); William M. Landes & Richard A. Posner, *An Empirical Analysis of the Patent Court*, 71 U. CHI. L. REV. 111, 111–12 (2003) (positing that a specialized patent court is more likely than a generalist court to take a strong stance on its subject matter because "interest groups that had a stake in patent policy would be bound to play a larger role in the appointment of the judges of such a court than they would in the case of the generalist federal courts").

[328] *See* LAWRENCE BAUM, SPECIALIZING THE COURTS 181, 204 (2011) (noting that corporate support played a key role in creation of the Federal Circuit); Paul R. Gugliuzza, *Rethinking Federal Circuit Jurisdiction*, 100 GEO. L.J. 1437, 1458 (2012); F.M. Scherer, *The Political Economy of Patent Policy Reform in the United States*, 7 J. TELECOMM. & HIGH TECH. L. 167, 190 (2009) (noting the strong support from corporate patent counsel).

> been responsible for the court's distinctive path in the past two decades.[329]

The legislation creating the Federal Circuit as well as the initial appointment of jurists from the CCPA implemented a mission of strengthening the patent system through statutory interpretation and evolution of non–statutory patent doctrines.[330] This mission has been reinforced through the emergence of a dedicated, well–funded bar and numerous patent–focused industry organizations.[331] While such organizations produce valuable research and education, it would be naive to think that the ecosystem surrounding patent adjudication did not promote the agenda of the most active and interested constituencies.[332]

Academic research finds that the Federal Circuit has strengthened patent protection through statutory interpretation and evolution of non–statutory patent doctrines. Multiple scholars have chronicled particular doctrinal patterns (such as formalism and textualism) that strengthen patent

[329] *See* Lawrence Baum, *Judicial Specialization, Litigant Influence, and Substantive Policy: The Court of Customs and Patent Appeals*, 11 LAW & SOC'Y REV. 823, 845–46 (1977).

[330] Gugliuzza, *supra* note 328, at 1458 (discussing strong industrial support for creating the U.S. Court of Appeals for the Federal Circuit); LANDES & POSNER, *supra* note 327, at 26–27 (noting that the Federal Circuit "has defined its mission as promoting technological progress by enlarging patent rights").

[331] The Federal Circuit Bar Association, founded in 1985, "unites the various groups who practice within the Circuit community, including the private and public sectors and litigators as well as agency and house counsel." *See Mission & Vision*, FED. CIRCUIT BAR ASS'N, https://fedcirbar.org/About-FCBA/Who-We-Are/Mission-Vision (last visited Oct. 8, 2017) The American Bar Association, intellectual property owners, pharmaceutical industry, and high technology industries have long had strong advocacy arms. The biotechnology and software industries have become increasingly active. Not all of these constituencies favor strong patent rights, which produced a more complex political dynamic during the past two decades and the lead–up to the America Invents Act. Nonetheless, the Federal Circuit has remained focused on strong patent rights and a robust patent system.

[332] *See* LANDES & POSNER, *supra* note 327, at 26–27 (suggesting that Federal Circuit favors increased demand for the services of its primary constituency, patent lawyers); *see generally* William N. Eskridge, Jr., *Politics Without Romance: Implications of Public Choice Theory for Statutory Interpretation*, 74 VA. L. REV. 275 (1988).

protection.[333] Empirical research finds that the Federal Circuit views patent holders more favorably than regional circuit courts.[334]

The Federal Circuit, however, has not always expanded the scope of patent protection. For example, the Federal Circuit's formalism led the court to cabin the doctrine of equivalents.[335] More recently, the Federal Circuit has reined in patent damage theories.[336] On balance, however, the Federal Circuit has favored broad patentability,[337] narrow limitations,[338] and robust appellate authority.[339]

These tendencies raise the concern that the Federal Circuit would favor a broader scope of copyright protection for computer software than regional circuit courts. The Federal Circuit's 2014 *Oracle v. Google* decision appears to bear this out. The Federal Circuit downplayed the legislative concern for ensuring that copyright protection for computer software does not extend to functional features, which is the province of patent protection. Moreover, the court read the Ninth Circuit's jurisprudence, particularly the *Sega*

[333] *See* Dreyfuss, *supra* note 326; Peter Lee, *Patent Law and the Two Cultures*, 120 Yale L.J. 2 (2010) (arguing that the Federal Circuit's formalism disengages from technology); Arti K. Rai, *Engaging Facts and Policy: A Multi-Institutional Approach to Patent System Reform*, 103 COLUM. L. REV. 1035, 1103–14 (2003) (suggesting that formalism might mask bias); John R. Thomas, *Formalism at the Federal Circuit*, 52 AM. U. L. REV. 771 (2003) (cautioning that the Federal Circuit should look beyond certainty and predictability in developing legal rules).

[334] *See* John R. Allison & Mark A. Lemley, *Empirical Evidence on the Validity of Litigated Patents*, 26 AIPLA Q.J. 185, 205–06 (1998); Matthew D. Henry & John L. Turner, *The Court of Appeals for the Federal Circuit's Impact on Patent Litigation*, 35 J. LEGAL STUD. 85, 114 (2006); Glynn S. Lunney, Jr., *Patent Law, the Federal Circuit, and the Supreme Court: A Quiet Revolution*, 11 SUP. CT. ECON. REV. 1, 15–16 (2004).

[335] *See* John R. Thomas, *Formalism at the Federal Circuit*, 52 AM. U. L. REV. 771, 772–75 (2003); *cf.* Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co., 535 U.S. 722, 738 (2002) (rejecting the complete bar to the doctrine of equivalents and noting that "we have consistently applied the doctrine [of equivalents] in a flexible way, not a rigid one").

[336] *See* VirnetX, Inc. v. Cisco Sys., Inc., 767 F.3d 1308 (Fed. Cir. 2014); LaserDynamics Inc. v. Quanta Comput., Inc., 694 F.3d 51 (Fed. Cir. 2012); Uniloc USA, Inc. v. Microsoft Corp., 632 F. 3d 1292 (Fed. Cir. 2011).

[337] *See, e.g.* Biosig Instruments, Inc. v. Nautilus, Inc., 715 F.3d 891 (Fed. Cir. 2014), *vacated,* Nautilus, Inc. v. Biosig Instruments, Inc., 134 S. Ct. 2120 (2014); MercExchange, L.L.C. v. eBay, Inc., 401 F.3d 1323 (Fed. Cir. 2005), *rev'd*, eBay, Inc. v. MercExchange, L.L.C., 547 U.S. 388 (2006); Teleflex Inc. v. KSR Int'l Co., 119 Fed. Appx. 282 (Fed. Cir. 2005), *rev'd*, KSR Int'l Co., v. Teleflex Inc., 550 U.S. 398 (2007); State St. Bank & Tr. Co. v. Signature Fin. Grp., Inc., 149 F.3d 1368 (Fed. Cir. 1998).

[338] *See, e.g.,* Madey v. Duke Univ., 307 F.3d 1351, 1360–61 (Fed. Cir. 2002) (construing the common law experimental use defense narrowly).

[339] *See* Teva Pharm. USA, Inc. v. Sandoz, Inc., 723 F.3d 1363 (Fed. Cir. 2013), *vacated*, 135 S. Ct. 831 (2015).

decision, narrowly. Furthermore, the court placed great emphasis on its own application of Ninth Circuit law in the *Atari Games* case.[340]

It is perhaps surprising that the Federal Circuit's formalism and experience with patent law did not push the court toward a narrower scope of copyright protection. As the panel recognized, the fair use doctrine is the most unpredictable doctrine in copyright law.[341] Furthermore, broad copyright protection for computer software impinges on patent law's primacy in promoting technological advance. *Baker v. Selden* preempts copyright protection of functional elements and methods of operation. Nonetheless, protecting intellectual property carried the day. It is reasonable to believe that like the First Circuit in *Lotus v. Borland*, the Ninth Circuit would have taken greater cognizance of the anticompetitive concerns of broad intellectual property protection. The legislative history of the Federal Courts Improvement Act highlighted the concern that a patent–centric specialty court might be less sensitive to antitrust policy.[342]

4. *Administrative Efficiency*

Administrative efficiency weighs against regional circuit jurisprudential integrity, federalism, and specialization–bias considerations in allocating appellate jurisdiction.[343] Various doctrines promote bundling causes of action arising out of the same transaction or occurrence to prevent piecemeal litigation.[344] There may be administrative efficiency reasons to bundle the appeal of those causes of action as well, but such efficiencies might be relatively modest and in tension with other jurisdictional considerations.

[340] *See Oracle II*, 750 F.3d at 1357, 1360, 1361, 1363, 1366, 1370 (citing Atari Games Corp. v. Nintendo of Am., Inc., 897 F.2d 1572 (Fed. Cir. 1990) as substantive copyright law authority).

[341] *See Oracle II*, 750 F.3d at 1372 (quoting Monge v. Maya Magazines, Inc., 688 F.3d 1164, 1170 (9th Cir. 2012) (quoting Dellar v. Samuel Goldwyn, Inc., 104 F.2d 661, 662 (2d Cir. 1939) (per curiam)).

[342] *See supra* notes 305–307 and accompanying text.

[343] Congress considered administrative efficiencies in establishing the Federal Circuit. *See* Daniel J. Meador, *An Appellate Court Dilemma and a Solution Through Subject Matter Organization*, 16 U. MICH. J.L. REFORM 471 (1983).

[344] *See* RESTATEMENT (SECOND) OF JUDGMENTS § 24(1) (AM. LAW INST. 1982) ("When a valid and final judgment rendered in an action extinguishes the plaintiff's claim . . . the claim extinguished includes all rights of the plaintiff to remedies against the defendant with respect to all or any part of the transaction, or series of connected transactions, out of which the action arose."); *see generally* CHARLES A. WRIGHT & ARTHUR R. MILLER, FEDERAL PRACTICE & PROCEDURE §§ 4401–09 (3d ed. 2002).

District judges have substantial flexibility in managing litigation, including staging and bifurcation.[345] Judge Alsup sensibly phased the copyright and patent causes of action in *Oracle v. Google*, with both sets of issues tried to the same jury. This will generally be sound case management due to the substantial differences between patent and copyright law.

Given that Oracle did not appeal dismissal of the patent causes of action, there is no administrative efficiency (or other) basis for the Federal Circuit, as opposed to the Ninth Circuit, to hear the *Oracle v. Google* appeal. Speculatively inferring what a Ninth Circuit panel would do could have been avoided. Moreover, appeal to the Ninth Circuit would have assured regional circuit jurisprudential integrity. Furthermore, it is not clear that dividing the case along patent and nonpatent causes of action where both sets of issues are appealed would add substantial complexity. The next Section explores these scenarios

C. REFINING APPELLATE INTELLECTUAL PROPERTY JURISDICTION

The challenge for jurists and policymakers is promoting faithful interpretation of regional circuit nonpatent jurisprudence without jeopardizing administrative efficiency. The development of a distinct line of Federal Circuit interpretation of regional circuit jurisprudence creates the potential for a new form of forum shopping. By pursuing patent claims with nonpatent causes of action, parties can opt into the Federal Circuit's interpretation of regional circuit law. That might be advantageous where an intellectual property owner seeks a more expansive interpretation of intellectual property protection. Moreover, because of the lack of error correction short of Supreme Court review, the jurisprudential structure of the federal appellate judiciary will be distorted. Contrary to Congress's intent, the Federal Circuit can effectively override regional circuits' nonpatent bodies of law.

This Section proposes restructuring appellate intellectual property jurisdiction along case management lines. Subsection 1 analyzes case management options for district courts. Subsection 2 analyzes appellate case management reforms.

1. District Court Case Management and Routing of Appellate Review

The core appellate jurisdiction problem traces back to the filing of a complaint asserting patent and non–patent causes of action. Once a plaintiff

[345] *See generally* PETER S. MENELL ET AL., PATENT CASE MANAGEMENT JUDICIAL GUIDE (3d ed. 2016).

files a complaint containing a patent cause of action, Section 1295(a)(1) of Title 28 confers exclusive appellate jurisdiction on the Federal Circuit. Yet Congress seeks to have regional circuit law (or possibly state law) apply to the nonpatent causes of action. By starting with district court case management, various opportunities to promote fidelity to regional circuit can be pursued while preserving the Federal Circuit's authority to interpret and apply federal patent law.

Although the patent and nonpatent issues may share a common nucleus of operative facts, the applicable legal standard may be sufficiently distinct that it makes sense to phase or bifurcate trial of the causes of action, as occurred in the *Oracle v. Google* case. To the extent that the district court keeps the trial and post–trial rulings separate, it is as if separate cases have been adjudicated.

If exclusively patent or nonpatent issues are appealed, jurisdictional integrity and federalism considerations favor having those issues resolved by the appellate tribunal with primary authority: the Federal Circuit for patent issues and the regional circuit court for the nonpatent issues. Thus, since there is no loss in administrative efficiency for cases in which patent issues are not appealed, the most obvious solution would be to vest jurisdiction over the appeal of the nonpatent issues with the regional appellate court. This could be accomplished by amending § 1295(a)(1) of Title 28 of the U.S. Code to exclude from the Federal Circuit's appellate jurisdiction cases that do not appeal issues arising under the Patent Act or Plant Variety Protection Act.

If both patent and nonpatent issues are appealed, the only difference would be that the case would be effectively divided into separate causes of action and the timing of appeals and remands could affect case management. But since the case was already phased or bifurcated, appellate bifurcation would be straightforward and not add significant additional administrative cost. The district court would retain jurisdiction and could adapt any further proceedings based on the outcome and timing of the parallel appellate processes. Section 1295(a) could be amended to provide for cases in which the nonpatent issues have been tried separately—whether through phasing or bifurcation—to fall within the appellate jurisdiction of the regional circuit court of appeals.

That leaves cases in which the patent and nonpatent issues have been litigated in a combined proceeding and are intertwined. This might occur, for example, in cases involving patent claims and antitrust counterclaims. Even in such scenarios, the appeal could be best handled by the regional circuit court if no patent issues are appealed. If patent issues or patent and interrelated, nonpatent issues are appealed, then the Federal Circuit has primacy in adjudicating the appeal.

2. *Appellate Jurisdiction Reforms*

Even where the Federal Circuit considers regional circuit law questions, there are several opportunities to improve fidelity to regional circuit jurisprudence. Section (a) considers implementation of a system analogous to the certification of state law questions to the highest state court tribunal. Section (b) considers adding a second tier of appellate review at the regional circuit level. Section (c) discusses ramifications for the Supreme Court's role.

a) Certification of Questions to Regional Circuit Courts

The federal courts have long dealt with the interpretation and application of other bodies of law. Following the Supreme Court's decision in *Erie Railroad Co. v. Tompkins*[346] largely eliminating general federal common law, federal courts have had to apply state law in diversity jurisdiction cases. Beginning in the 1960s, most states have afforded federal courts the option of certifying questions of state law to the highest court in the state.[347] The federal court can go directly to the highest state court to resolve difficult interpretive questions. This process, however, is at the discretion of the federal court.

Although Congress could authorize an analogous process for the Federal Circuit to certify complex questions of regional circuit law to the regional circuit court, such a process would be unduly cumbersome. Unlike the highest court in a state, regional circuits typically sit in panels smaller than the full bench. En banc review is a relatively infrequent process. It is also relatively complicated to organize and it can take a long time to render decisions on complex issues. Furthermore, it might be difficult to boil down a question of regional circuit copyright law or other nonpatent issue to a clear question that can easily be applied. A better approach would be to develop a mechanism for direct review of the Federal Circuit's nonpatent issues at the regional circuit level.

b) Regional Circuit Review

Even if the nonpatent issues cannot be separated from the patent issues prior to the first appeal level, Congress could provide for Federal Circuit

[346] Erie R.R. Co. v. Tompkins, 304 U.S. 64 (1938).

[347] *See* Rebecca A. Cochran, *Federal Court Certification of Questions of State Law to State Courts: A Theoretical and Empirical Study*, 29 J. LEGIS. 157, 159 n.13 (2003).

interpretations of regional circuit review to create an additional layer of appellate review by a regional circuit panel and/or at the en banc level. Thus, Congress could provide for an optional second level of appellate review. A party that believed that the Federal Circuit has misinterpreted or misapplied regional circuit jurisprudence could challenge that decision through a second–panel review within the regional circuit. Alternatively, a party challenging the Federal Circuit's interpretation of regional circuit law could file an en banc petition in the regional circuit.

These approaches provide a sensible and balanced solution to the regional circuit jurisprudential integrity and forum shopping problems while avoiding undue administrative costs. Given the Supreme Court's severe capacity constraints[348] and disinclination to consider interlocutory appeals[349] and intra–circuit splits,[350] providing litigants a regional circuit review option could provide a valuable secondary screen to ensure fidelity to regional circuit authority. It could also avoid the additional costs from Federal Circuit remands on unnecessary issues. On the cost side of the equation, an additional appellate review would add further time to resolving disputes. But in cases like *Oracle v. Google*, such an option would have potentially avoided a costly second trial and would likely have provided a clear answer to core questions about API copyrightability in the Ninth Circuit.

A regional circuit appeal process would also conserve Supreme Court resources. When the Federal Circuit misinterprets regional circuit law, it effectively creates an intra–circuit split. If Google had the option to pursue regional circuit review, it could have avoided filing its interlocutory certiorari petition. The copyright issues would have found their path within the regional circuit process. And only if one of the parties could allege an inter–circuit split would the case become ripe for Supreme Court review.

Whether to limit the second–level, regional appeal to the panel or en banc level would depend on an analysis of process costs and delay. Limiting a Federal Circuit litigant only to regional circuit en banc review of nonpatent issues would save resources, but would likely result in more Federal Circuit mutation of regional circuit law.

[348] *See* Ryan J. Owens & David A. Simon, *Explaining the Supreme Court's Shrinking Docket*, 53 WM. & MARY L. REV. 1219 (2012); Kenneth W. Starr, *The Supreme Court and Its Shrinking Docket: The Ghost of William Howard Taft*, 90 MINN. L. REV. 1363 (2006); HRUSKA COMMISSION REPORT, *supra* note 5, at 209–14.

[349] *See* SUP. CT. R. 11.

[350] *See* SUP. CT. R. 10; Textile Mills Sec. Corp. v. Comm'r, 314 U.S. 326, 344–45 (1941).

c) Supreme Court Review

If Congress does not act to route appeals of separable nonpatent issues to regional circuits and provide an additional layer of circuit review, either panel or en banc, for nonpatent issues remaining after Federal Circuit review, then the Supreme Court should expand its certiorari criteria to consider Federal Circuit misinterpretation of regional circuit law. As the *Oracle v. Google* litigation highlights, it is possible that the copyrightability of APIs might never be ripe for Supreme Court review. The detour into a fair use trial has submerged the API copyrightability issue, which is of great importance to a substantial portion of the software industry.

VI. CONCLUSIONS

The Federal Circuit's exclusive jurisdiction over patent appeals, in conjunction with its questionable interpretation of Ninth Circuit copyright law, has produced a new class of forum shopping. By combining a patent cause of action with a software copyright cause of action in a district court filing anywhere within the boundaries of the Ninth Circuit, a plaintiff can opt for a far more expansive version of copyright protection than they could obtain if they only pursued the copyright cause of action. Congress specifically warned against the extension of the Federal Circuit's exclusive jurisdiction over patent appeals to nonpatent causes of action governed by regional circuit law in establishing the Federal Circuit.

Fortunately, some fairly straightforward adjustments to the Federal Circuit's jurisdiction can nip this problem in the bud. Congress can largely rectify the problem by providing for nonpatent issues resolved in separable proceedings to be appealed directly to the regional circuit court. Furthermore, Congress can address the problem presented by intertwined patent and nonpatent issues by providing for a second level of appellate review of the nonpatent issues in the regional circuit.