Nos. 17-1118, -1202

IN THE UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

ORACLE AMERICA, INC.,

Plaintiff-Appellant,

v.

GOOGLE LLC,

Defendant-Cross-Appellant.

Appeal from the United States District Court for the Northern District of California, Case No. 3:10-cv-03561-WHA, Hon. William H. Alsup

PETITION FOR REHEARING *EN BANC*

Robert A. Van Nest
Christa M. Anderson
Reid P. Mullen
KEKER, VAN NEST & PETERS LLP
633 Battery Street
San Francisco, CA 94111
Telephone: (415) 391-5400

Renny Hwang
GOOGLE LLC
1600 Amphitheatre Parkway
Mountain View, CA 94043
Telephone: (650) 253-2551

Daryl L. Joseffer
Bruce W. Baber
Anne M. Voigts
Marisa C. Maleck
KING & SPALDING LLP
1700 Pennsylvania Ave. NW
Washington, D.C. 20006
Telephone: (202) 737-0500
djoseffer@kslaw.com

Counsel for Defendant-Appellee/Cross-Appellant Google LLC

May 29, 2018

CERTIFICATE OF INTEREST

Counsel for Cross-Appellant Google LLC certifies the following:

1. The full name of the party represented by me is Google LLC.

2. The name of the real party in interest is Google LLC.

3. Google LLC is a wholly owned subsidiary of XXVI Holdings Inc., which is a wholly owned subsidiary of Alphabet Inc., a publicly traded company (NASDAQ: GOOG, GOOGL). No publicly traded company owns 10% or more of Alphabet Inc.'s stock.

4. The names of all firms and the partners or associates that appeared for the party now represented by me in the trial court or are expected to appear in this court and who are not already listed on the docket for this case are:

   KING & SPALDING LLP: Christopher C. Carnaval; Geoffrey M. Ezgar; Truman H. Fenton; Mark H. Francis; Robert F. Perry; Cheryl A. Sabnis; Steven T. Snyder; Scott T. Weingaertner; Joseph R. Wetzel; and Donald F. Zimmer, Jr.

   KEKER & VAN NEST LLP: Edward Andrew Bayley; Elizabeth Ann Egan; Sarah Brienne Faulkner; Matthias Andreas Kamber; Maya Beth Karwande; Kate Ellis Lazarus; Eugene Morris Paige; Daniel Edward Purcell; Steven Ragland; and David Zimmer.

   GREENBERG TRAURIG LLP: Ian Ballon; Valerie Wing Ho; Wendy Michelle Mantell; Dana K. Powers; and Luis Villa, IV.

5. I am not aware of a case pending in this Court or another court or agency that will directly affect or be directly affected by this Court's decision in this case.

May 29, 2018

/s/ *Daryl L. Joseffer*
Daryl L. Joseffer

TABLE OF CONTENTS

TABLE OF AUTHORITIES

CASES

STATUTES

CONSTITUTIONAL PROVISIONS

LEGISLATIVE MATERIAL

OTHER AUTHORITIES

STATEMENT OF COUNSEL REQUIRED BY FED. CIR. R. 35(B)

Based on my professional judgment, I believe the panel decisions are contrary to at least the following decisions: *Campbell v. Acuff-Rose Music*, 510 U.S. 569 (1994); *Harper & Row Publishers v. Nation Enters.*, 471 U.S. 539 (1985); *Sony Computer Entm't v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000); *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided court*, 516 U.S. 233 (1996); *Sega Enters. v. Accolade*, 977 F.2d 1510 (9th Cir. 1992).

Also based on my professional judgment, I believe this petition presents novel questions of exceptional importance:

1. Whether application programming interface ("API") declarations—which are designed to invoke pre-written functions and methods of software—are systems or methods of operation and thus not entitled to copyright protection.[1]

[1] Oracle accused Google of copying some of the declarations and the structure, sequence, and organization ("SSO") of certain Java API packages. The SSO is defined by the declarations. *See Oracle I*, 750 F.3d at 1351. This petition's discussion of the declarations applies equally to the SSO.

2. Whether use of API declarations, but not implementing code, in a new and different context is protected by the fair-use doctrine.

May 29, 2018

/s/ *Daryl L. Joseffer*
Daryl L. Joseffer

INTRODUCTION

Software developers have long understood—and the law has long recognized—that they are free to use existing computer software interfaces. Especially in this industry, innovation depends on the ability to interact with and improve on what has come before.

The panel's decisions in this high-profile case upend this settled understanding and throw a devastating one-two punch at the computer software industry. In its first decision, the panel held that all aspects of the API for the Java programming language are entitled to copyright protection, including the "method headers" or "declarations" that serve as commands for invoking pre-written Java functions and methods. *Oracle I*, 750 F.3d at 1376. It reached this holding despite the Copyright Act's express exclusion of "systems" and "methods of operation" from copyright protection. Google petitioned for certiorari, and the Supreme Court called for the Solicitor General's views. The government recommended against granting certiorari based in part on the case's interlocutory posture, and stressed that Google's "important concerns about the effects that enforcing [Oracle's] copyright could have on software development . . . are better addressed through [Google's]

fair-use defense, which will be considered on remand." U.S. *Amicus* Br., No. 14-401, at 10, 22.

On remand, the jury found that Google's use was fair after hearing from 29 witnesses and viewing hundreds of exhibits during a two-week trial. The district court denied Oracle's motion for judgment as a matter of law in a thorough opinion. But the panel reversed again, this time holding, contrary to its first decision, that Google's independent implementation of the Java API declarations was not a fair use as a matter of law—even though no party challenged the jury instructions, which were drawn directly from the panel's first opinion. That reversal was extraordinary: of the more than 300 decisions on fair use, we are aware of only one other decision overturning a jury verdict on fair use, and that district court decision is now on appeal. *See Corbello v. DeVito*, 262 F. Supp. 3d 1056 (D. Nev. 2017), *appeal pending*, No. 17-16337 (9th Cir.); Pamela Samuelson (@PamelaSamuelson), TWITTER (Mar. 27, 2018, 5:46 PM), https://twitter.com/PamelaSamuelson/status/978795513809010688.

This case was a most unlikely candidate for such unusual treatment. Google's use of the Java API declarations is a quintessential

example of transforming a portion of existing software for desktops—here, 11,500 lines out of about 5 million lines of the copyrighted work—in a new context: smartphones. Although software can generally be protected by copyright, API declarations are different. The very purpose of an API is to create a set of words—the declarations—that a developer can copy (and must copy exactly) into a program to invoke pre-written functions and methods. If the panel's decision is allowed to stand, it is hard to see how any adaptation of any element of computer software to a new context could ever qualify as fair use.

FACTUAL BACKGROUND

"Oracle's predecessor, Sun Microsystems, Inc. ('Sun'), developed the Java platform for computer programming." *Oracle II*, 886 F.3d at 1186. "The Java platform is software used to write and run programs in the Java programming language." *Id.* It includes the Java API, "a collection of 'pre-written Java source code programs.'" *Id.*

Each pre-written program, or "method," "performs a specific function, sparing a programmer the need to write Java code from scratch to perform that function." *Id.* Each method has (i) a method header or declaration, which a programmer can use to cause a computer

to run code; and (ii) the method body or implementing code, which "gives the computer step-by-step instructions to carry out the declared function." *Id.*

Sun encouraged programmers to learn and use Java by making the language free for all to use and by touting the openness and accessibility of Java's API, as well as the ease of programming in Java made possible by the pre-written functions and methods for common operations contained in the API. Google Br. 9–10; *Oracle II*, 886 F.3d at 1203. Sun's former CEO testified that Sun marketed the API "as free and open," and "never" considered the declarations to be "proprietary." Google Br. 45.

That was not altruistic. Sun encouraged millions of programmers to adopt the efficient programming made possible by APIs. *Id.* at 10. Sun hoped the resulting ubiquity of Java-proficient programmers would drive sales of Sun hardware and other services (as it did). *Id.* at 11. Just as other industry participants copied the necessary declarations from the Java APIs, but wrote their own implementing code, Sun too copied others' APIs in building its own products. *Id.*

Google is the lead developer of Android, an operating system designed for modern smartphones. To allow programmers to use their Java language skills (including familiar declarations to access pre-written functions) in creating applications for Android, Google used the declarations from 37 Java API packages. *See Oracle II*, 886 F.3d at 1187. The jury heard evidence that, unlike implementing code, the declarations are not highly creative. Google Br. 29.

Google did not use any of the implementing code in the API packages, which were designed for desktops and servers only, and were unsuited for the mobile environment's fundamentally different constraints. Google Br. 15–16. Google used only the subset of Java declarations from the APIs it deemed necessary and appropriate for mobile devices, and created its own implementing code. *Id*. Google also created more than 100 totally new API packages for additional functions needed in the mobile environment. *Id*. In total, Android contained more than 15 million relevant lines of code, including about 11,500 lines of Java API declarations (about 0.4% of the Java platform and 0.08% of Android). *Oracle II*, 886 F.3d at 1187; *see also* Google Br. 17.

Sun and Oracle both celebrated Android's use of Java. Sun's CEO publicly offered "heartfelt congratulations" to Google, stating that Google had "strapped another set of rockets to the [Java] community's momentum." Google Br. 18. Oracle's CEO also embraced Android after Oracle acquired Sun. *Id.* at 19. But Oracle later claimed that Android infringed Java copyrights.

ARGUMENT

I. Methods Of Operation Embodied In Computer Programs Are Not Entitled To Copyright Protection.

Section 102(b) of the Copyright Act specifies that "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b). That provision codifies the Supreme Court's holding in *Baker v. Selden*, 101 U.S. 99, 105 (1879), that systems and methods of operation are the subject of patent law (and its 20-year term) and not copyright law (with its term of 95 years or longer).

The panel held that, notwithstanding its plain language, Section 102(b) does *not* exclude systems or methods of operation from copyright

protection, and that all elements of an original work are "entitled to copyright protection as long as the author had multiple ways to express the underlying idea." *Oracle I,* 750 F.3d at 1367. In its view, Section 102(b) indicates only "that certain expressions are subject to greater scrutiny." *Id.* at 1357.

The panel did not identify the source of this "greater scrutiny" test. Nor did it apply greater scrutiny; it simply held that because Sun could have written different declarations, they were copyrightable. In contrast, the Supreme Court determined nearly 30 years ago that, by its plain language, Section 102(b) "identifies specifically those elements of a work for which copyright is not available." *Feist Publ'ns, Inc. v. Rural Tel. Serv.*, 499 U.S. 340, 356 (1991).

Other appellate courts have declined to rewrite that plain statutory text. In *Lotus*, for example, the First Circuit held that because a spreadsheet program's "menu command hierarchy provides the means by which users control and operate" the program, the command hierarchy is not entitled to copyright protection—even though "Lotus developers could have designed [it] differently." 49 F.3d at

815–16. The Supreme Court granted certiorari and deadlocked, affirming by an equally divided court. 516 U.S. 233.

The Ninth Circuit has held that "aspects" of a computer program that constitute "functional requirements for compatibility" with other programs are "not protected by copyright" under Section 102(b) (and that other aspects "'necessary' to gain access to the functional elements" may also be copied under the fair-use doctrine). *See Sega*, 977 F.2d at 1522; *Sony*, 203 F.3d at 603. After *Oracle I*, the Ninth Circuit confirmed that it does not matter whether there is more than one way to design a system or method: "the possibility of attaining a particular end through multiple different methods does not render the uncopyrightable a proper subject of copyright." *Bikram's Yoga College of India v. Evolation Yoga*, 803 F.3d 1032, 1042 (9th Cir. 2015).

The panel's rewriting of the statute was dispositive because the Java API declarations are the system or method of operating the pre-written programs. They are "commands to instruct a computer to carry out desired operations." *Oracle I*, 750 F.3d at 1367. If the declarations were changed, the familiar shorthand access to pre-written functions and methods would not work. Google Br. 8–9. When specific computer

code must be used to operate computer programs, that code is necessarily part of the system or method of operating those programs. *See, e.g., Lotus*, 49 F.3d at 817–18.

The merger doctrine further confirms the panel's error. That doctrine is rooted in the idea/expression dichotomy, which generally excludes ideas but not expression from copyright protection. If there is only one practical way to express an idea, that expression essentially merges with the idea and is not copyrightable. *E.g.*, *Lexmark Int'l v. Static Control Components*, 387 F.3d 522, 534–35 (6th Cir. 2004). That is the case here. If Google did not use the API's declarations exactly as defined in Java, programmers could not have used the familiar shorthand commands.

II. Google's Use Was Not Unfair As A Matter Of Law.

"From the infancy of copyright protection, some opportunity for fair use of copyrighted materials has been thought necessary to fulfill copyright's very purpose, '[t]o promote the Progress of Science and useful Arts. . . .'" *Campbell*, 510 U.S. at 575 (citing U.S. Const., Art. I, § 8, cl. 8). Fair use is especially important for computer code, which is "essentially utilitarian" and, being largely functional, "receives only

weak protection" under the Copyright Act. *Sega*, 977 F.2d at 1525, 1527. As the Ninth Circuit has held, the "fair use doctrine preserves public access to the . . . functional elements embedded in copyrighted computer software programs," *Sony*, 203 F.3d at 603, to encourage "the dissemination of other creative works" that build on those functional elements. *Sega*, 977 F.2d at 1523. The panel erred in reversing the jury's verdict.

As an initial matter, the panel applied the wrong standard of review. Fair use is a mixed question of law and fact. *Harper & Row*, 471 U.S. at 560. A factfinder's conclusion on a mixed question must be reviewed deferentially where, as here, the question requires a case-specific, fact-intensive analysis. *U.S. Bank Nat. Ass'n ex rel. CWCapital Asset Mgmt. v. Vill. at Lakeridge*, 138 S. Ct. 960, 965 (2018). The panel reviewed the jury's ultimate determination of fair use *de novo* on the theory that "the fair use determination definitely does not 'resist generalization'" based on an individual case's facts. *Oracle II*, 886 F.3d at 1193 (quoting *U.S. Bank*, 138 S. Ct. at 966). But the Supreme Court has repeatedly stressed that each fair use case must be decided "on its own facts," *Harper & Row*, 471 U.S. at 560 (quoting H.R. Rep. No. 94-

1476), and that rigid, "bright-line rules" are therefore improper. *Campbell*, 510 U.S. at 577; *accord Sony Corp. v. Universal City Studios*, 464 U.S. 417, 448 & n.31 (1984).

The panel nonetheless adopted as fact a large number of disputed Oracle assertions contrary to the jury's presumed findings. For example, the panel found that Android used the API declarations in the same context as the Java platform, and that Android caused actual market harm to Java, because "Java SE APIs were in smartphones before Android entered the market." *Oracle II*, 886 F.3d at 1201. But the panel failed to consider the evidence that these early phones were nothing like modern smartphones; these early phones did not even use the copyrighted work, Java SE; and even Sun's own CEO did not believe that Java SE was in the smartphone market. *See* Google Br. 13–14, 60–62.

The panel also concluded that Android harmed the potential market for Java because the "undisputed evidence showed . . . that Oracle intended to license Java SE in smartphones." 886 F.3d at 1210. But the jury heard evidence from Sun's own CEO that Sun marketed

the Java APIs "as free and open" for all to use without a license. Google Br. 12, 45.

The panel asserted that the parties had stipulated that only 170 lines of code were necessary to write in the Java language. 886 F.3d at 1206. But the stipulation was "without prejudice to evidence that other additional declaring code beyond those lines . . . either was or was not necessary for the use of the Java programming language." Appx51447–51448.[2]

In addition to applying the wrong standard of review, the panel committed further errors of law:

[2] Those are but a few, non-exclusive examples of the panel adopting Oracle's assertions on disputed questions. Others include: Android was a "mere change in format" for the Java APIs from desktop and laptop computers to smartphones and tablets, 886 F.3d at 1202; the declaring code that Google copied was "qualitatively significant," *id.* at 1205–06; "Oracle charges a licensing fee to those who want to use the APIs in a competing platform or embed them in an electronic device," *id.* at 1187; "Oracle licensed . . . Java SE for mobile devices," *id.* at 1187; "Android was used as a substitute for Java SE and had a direct market impact," *id.* at 1209; "Android was devastating to [Oracle's] licensing strategy" for J2SE, *id.* at 1187; Amazon switched to Android for the Kindle Fire and then used the existence of Android to leverage a steep discount from Oracle on the next generation Kindle, *id.* at 1187–88; and SavaJe and Danger were smartphones that used J2SE. *Id.* at 1201.

1. The Panel Misapplied The Law As To The Statutory Fair-Use Factors.

"[T]he goal of copyright, to promote science and the arts, is generally furthered by the creation of transformative works." *Campbell*, 510 U.S. at 579. Thus, the foundational issue in most fair use cases, including this one, is "whether and to what extent the new work is 'transformative.'" *Id.*

The panel concluded that Google's use was not transformative primarily because it did not change the copied declarations and those declarations serve "an identical function and purpose" in Android. *Oracle II*, 886 F.3d at 1202. Under that approach, virtually no use of any element of a computer program could ever be transformative. The copied elements invariably would perform the same function in the new work because they were designed to perform that specific function. That has never been the law, as confirmed by long-standing precedents holding that copying of computer programs' functional elements was a fair use "notwithstanding the similarity of uses and functions" in competing products. *Sony*, 203 F.3d at 606, 608; *see Sega*, 977 F.2d at 1521–22.

The panel also erred by focusing on transformation of the declarations, as opposed to the works (the Java 2 SE platform and Android) as a whole. The court analyzed, for example, "the purpose of the API packages." *Oracle II*, 886 F.3d at 1199. But the question is whether "the new work" *as a whole* is transformative, *Campbell*, 510 U.S. at 579, not whether the copied portion has been transformed. In *Campbell*, for example, the Court analyzed whether a commercial parody of a song transformed the song, not whether some of the lines in each song were the same (they were). *Id.* at 583–85. A news program's video montage was transformative even though it included an unaltered, newsworthy video clip. *L.A. News Serv. v. CBS Broad.*, 305 F.3d 924, 939–40 (9th Cir. 2002). An artistic video was transformative even though it included an unaltered piece of art. *Seltzer v. Green Day*, 725 F.3d 1170, 1176–78 (9th Cir. 2013).

In all of those cases, the new work transformed the original work because of what it added to that work. In the panel's view, however, it is "irrelevant" that "Google wrote its own implementing code." *Oracle II*, 886 F.3d at 1201. That only underscores how far the panel strayed from the correct legal analysis. What Google added to the Java API

declarations as part of its new work is not only relevant, but *essential* to assessing the extent to which the new work transformed the old. Google selected particular declarations coupled "with new implementing code adapted to the constrained operating environment of mobile smartphone devices with small batteries" and "combined with brand new methods, classes, and packages written by Google for the mobile smartphone platform." Appx42; *see also* Google Br. 15–17.

By any measure, that is far more transformative than other uses found to be transformative. *See, e.g., Fox News Network v. TVEyes*, 883 F.3d 169, 172–73, 176–78 (2d Cir. 2018) (service that recorded entire televised program segments, and enabled consumers to "easily locate and view" those segments, was "at least somewhat transformative"—even though the unaltered segments were used "for the same purpose" as the originals); *see also Swatch Group Mgt. Servs. v. Bloomberg L.P.*, 756 F.3d 73, 84–85 (2d Cir. 2014) (two works "had different messages and purposes" "notwithstanding that the data . . . was identical" in both).

The panel's assessment of transformativeness was crucial to its finding of no fair use. Congress has set forth four *non-exclusive* factors

for juries and courts to consider. *See* 17 U.S.C. § 107; *Campbell*, 510 U.S. at 577. After recognizing that transformativeness is "central" to the first factor (the purpose and character of the use), the panel concluded that this factor "weigh[s] heavily against a finding of fair use." *Oracle II,* 886 F.3d at 1198, 1210. It also found that another factor (amount and substantiality of the portion used) was "at best, neutral" and "arguably weighs against" fair use, in large part because it had "found . . . Google's use . . . not transformative." *Id.* at 1207, 1210. The panel's legal error on transformativeness was, therefore, undeniably important to its conclusion (especially considering that it found that another factor, the nature of the copyrighted work, weighs in favor of fair use).

2. The Panel Ignored Other Highly Probative Considerations.

Even though the Supreme Court has repeatedly stressed that fair use is a flexible analysis based on the entire "universe of relevant evidence," *Campbell*, 510 U.S. at 584, the panel ignored a crucial consideration: that "fair use is appropriate where a 'reasonable copyright owner' would have consented to the use, i.e., where the 'custom or public policy' at the time would have defined the use as

reasonable." *Wall Data v. L.A. Cty. Sheriff's Dep't*, 447 F.3d 769, 778 (9th Cir. 2006). The jury heard uncontradicted, compelling testimony from Sun's former CEO and CTO that Sun's business model was to make the declarations free for anyone to use, and that Google's use of those declarations furthered that model. Google Br. 9–12, 19. In other words, not only would a reasonable copyright owner have consented to the use, Sun actually did. The jury also heard that it was industry custom to copy and use others' APIs. *Id.* at 46–47. Although this evidence featured prominently at trial, and the district court emphasized its significance, Appx39, the panel ignored it.

III. This Case Is Exceptionally Important.

This is the "copyright lawsuit of the decade," and if the panel's rulings are not corrected, their reverberations will be felt for decades. Anandashankar Mazumdar, *Oracle Victory Stirs Uncertainties in Software Copyright,* Bloomberg Law, May 10, 2018, https://biglawbusiness.com/oracle-victory-stirs-uncertainties-in-software-copyright/. The panel's conclusion that a firmly established, widely practiced method of designing computer software violates copyright law presents exceptionally important questions that warrant review. The

software industry needs a total overhaul to avoid crippling liability and faces huge new barriers to innovation. There will be acute pressure to "settl[e] early" suits over prior versions of products. *Id.*

Had the panel's decisions been the law at the inception of the Internet age, early computer companies could have blocked vast amounts of technological development by claiming 95-year copyright monopolies over the basic building blocks of computer design and programming. The large number of panel-stage amicus briefs (19 during the second appeal alone) and the extensive critical commentary on this Court's decisions confirm their exceptional importance. *See, e.g.*, Urmika Devi Shah, *Decision in Oracle v. Google Fair Use Case Could Hinder Innovation in Software Development*, Mozilla, Apr. 17, 2018, https://blog.mozilla.org/netpolicy/2018/04/17/decision-in-oracle-v-google-fair-use-case-could-hinder-innovation-in-software-development/ ("Because of this ruling, copyright law today is now at odds with how software is developed."); Van Lindberg, *The Copyrightability Of APIs In The Land Of OpenStack*, Rackspace, May 16, 2014, https://blog.rackspace.com/the-copyrightability-of-apis-in-the-land-of-openstack ("This move overturns the expectations of businesses and developers

and is likely to negatively impact how they leverage APIs going forward.").

Although the panel purported to apply Ninth Circuit law, this is now a Federal Circuit problem. "[G]iven the proliferation of software patents, a company with a widely used set of APIs could very likely pursue both patent and copyright causes of action in the same litigation, thereby bringing the Federal Circuit's exclusive jurisdiction over patent cases into play, even where patent issues are not appealed." Peter S. Menell, *Rise of the API Copyright Dead?*, 31 Harv. J. L. & Tech. 305, 414 (2018). That has already begun to happen over the last few years, as copyright plaintiffs have increasingly asserted patent claims as well. *See, e.g.*, *Synopsys v. ATopTech*, No. 13-CV-02965 (N.D. Cal. filed Jun. 26, 2013), *petition denied,* No. 14-124 (Fed. Cir. 2014); *Cisco Systems v. Arista Networks*, No. 14-CV-05344 (N.D. Cal. filed Dec. 5, 2014), *appeal pending*, No. 17-2145 (Fed. Cir.).

CONCLUSION

The Court should grant rehearing *en banc*.

Respectfully submitted,

/s/ *Daryl L. Joseffer*

Robert A. Van Nest
Christa M. Anderson
Reid P. Mullen
KEKER, VAN NEST & PETERS LLP
633 Battery Street
San Francisco, CA 94111
Telephone: (415) 391-5400

Renny Hwang
GOOGLE LLC
1600 Amphitheatre Parkway
Mountain View, CA 94043
Telephone: (650) 253-2551

Daryl L. Joseffer
Bruce W. Baber
Anne M. Voigts
Marisa C. Maleck
KING & SPALDING LLP
1700 Pennsylvania Ave. NW
Washington, D.C. 20006
Telephone: (202) 737-0500
djoseffer@kslaw.com

Counsel for Defendant-Appellee/Cross-Appellant Google LLC

May 29, 2018

ADDENDUM

> from a source of genetically identical cells share the same nucleus, but are not strictly identical as they are derived from different oocytes. The significance of this different origin is not clear, but may affect commercial traits." The Specification cautions further that "[i]t remains . . . to consider whether it is possible or necessary in specific situations to consider the selection of oocytes." Thus . . . the Specification does not disclose any systematic differences in the clones that arise from the capture of the recipient oocyte.

J.A. 12 (third, fourth, and fifth alterations in original) (citations omitted). There is nothing in the claims, or even in the specification, that suggests that the clones are distinct in any relevant way from the donor animals of which they are copies. The clones are defined in terms of the identity of their nuclear DNA to that of the donor mammals. To be clear, having the same nuclear DNA as the donor mammal may not necessarily result in patent ineligibility in every case. Here, however, the claims do not describe clones that have markedly different characteristics from the donor animals of which they are copies.

[9] Finally, Roslin argues that its clones are patent eligible because they are time-delayed versions of their donor mammals, and therefore different from their original mammals. But this distinction cannot confer patentability. As the Board noted, "[t]he difficulty with the time-delayed characteristic is that it is true of any copy of an original." J.A. 18. Thus, we affirm the Board's finding that Roslin's clones are unpatentable subject matter under § 101.

AFFIRMED.

ORACLE AMERICA, INC., Plaintiff–Appellant,

v.

GOOGLE INC., Defendant–Cross–Appellant.

Nos. 2013–1021, 2013–1022.

United States Court of Appeals, Federal Circuit.

May 9, 2014.

Background: Computer technology developer brought action alleging that search engine operator's operating system for mobile devices infringed its patents and copyrights relating to software programming language. The United States District Court for the Northern District of California, William Alsup, J., 847 F.Supp.2d 1178, granted in part and denied in part operator's motion in limine to exclude portions of developer's expert damages report, entered judgment in favor of operator on patent claims, and entered judgment in favor of developer in part and operator in part on copyright claims. Both parties appealed.

Holdings: The Court of Appeals, O'Malley, Circuit Judge, held that:

(1) idea contained in developer's computer code did not merge with its expression;

(2) developer's declaring source code contained protectable expression that was entitled to copyright protection;

(3) a set of commands to instruct a computer to carry out desired operations may contain expression that is eligible for copyright protection;

(4) developer's packages of application programming interfaces were not precluded from copyright protection just because they also performed functions; and

(5) operator's copying of developer's software was not de minimus.

Affirmed in part, reversed in part, and remanded.

1. Courts ⚿96(7)

When the questions on appeal involve law and precedent on subjects not exclusively assigned to the Federal Circuit, the court applies the law which would be applied by the regional circuit.

2. Federal Courts ⚿3943

Copyright issues are not exclusively assigned to the Federal Circuit.

3. Federal Courts ⚿3635

Whether particular expression is protected by copyright law is subject to de novo review by the Court of Appeals.

4. Copyrights and Intellectual Property ⚿10.4

Computer programs can be subject to copyright protection under the Copyright Act as "literary works." 17 U.S.C.A. § 101.

5. Copyrights and Intellectual Property ⚿12(1)

The originality requirement for copyright protection is not particularly stringent. 17 U.S.C.A. § 102(a).

6. Copyrights and Intellectual Property ⚿12(1)

"Original," as the term is used in copyright, means only that the work was independently created by the author and that it possesses at least some minimal degree of creativity. 17 U.S.C.A. § 102(a).

See publication Words and Phrases for other judicial constructions and definitions.

7. Copyrights and Intellectual Property ⚿4.5

Copyright protection extends only to the expression of an idea, not to the underlying idea itself. 17 U.S.C.A. § 102(b).

8. Copyrights and Intellectual Property ⚿10.4

The literal elements of a computer program, i.e., the source code and object code, may be entitled to copyright protection.

9. Copyrights and Intellectual Property ⚿10.4

The non-literal components of a computer program, including the program's sequence, structure, organization, and user interface, may be protected by copyright when they constitute an expression of an idea, rather than an idea itself. 17 U.S.C.A. § 102(b).

10. Copyrights and Intellectual Property ⚿53(1)

In the copyright context, "literal copying" is verbatim copying of original expression, while "non-literal copying" is paraphrased or loosely paraphrased rather than word for word.

See publication Words and Phrases for other judicial constructions and definitions.

11. Copyrights and Intellectual Property ⚿4, 4.5

In assessing copyrightability, a district court is required to ferret out apparent expressive aspects of a work and then separate protectable expression from unprotectable ideas, facts, processes, and methods of operation. 17 U.S.C.A. § 102.

12. Copyrights and Intellectual Property ⚿10.4

The abstraction-filtration-comparison test for assessing whether the non-literal elements of a computer program constitute expression protectable by copyright eschews bright line approaches and requires a more nuanced assessment of the particular program at issue in order to determine what expression is protectable and infringed. 17 U.S.C.A. § 102(a).

13. Copyrights and Intellectual Property ⚿10.4

In the abstraction step of the abstraction-filtration-comparison test for assessing whether the non-literal elements of a computer program constitute expression protectable by copyright, the court first breaks down the allegedly infringed program into its constituent structural parts, in the filtration step, the court sifts out all non-protectable material, including ideas and expression that is necessarily incidental to those ideas, and in the final step, the court compares the remaining creative expression with the allegedly infringing program. 17 U.S.C.A. § 102(a).

14. Copyrights and Intellectual Property ⚿10.4

The full analysis of the abstraction-filtration-comparison test for assessing whether the non-literal elements of a computer program constitute expression protectable by copyright only applies where a copyright owner alleges infringement of the non-literal aspects of its work; where admitted literal copying of a discrete, easily-conceptualized portion of a work is at issue, a court need not perform a complete abstraction-filtration-comparison analysis and may focus the protectability analysis on the filtration stage, with attendant reference to standard copyright principles. 17 U.S.C.A. § 102(a).

15. Copyrights and Intellectual Property ⚿10.4

In the filtration step of the test for assessing whether the non-literal elements of a computer program constitute expression protectable by copyright, the court is first to assess whether the expression is original to the programmer or author; the court must then determine whether the particular inclusion of any level of abstraction is dictated by considerations of efficiency, required by factors already external to the program itself, or taken from the public domain—all of which would render the expression unprotectable. 17 U.S.C.A. § 102(a).

16. Copyrights and Intellectual Property ⚿4.5, 12(2)

While questions regarding originality are considered questions of copyrightability, concepts of merger and scenes a faire are affirmative defenses to claims of infringement. 17 U.S.C.A. § 102(b).

17. Copyrights and Intellectual Property ⚿88

Computer technology developer did not waive its arguments based on search engine operator's literal copying of declaring source code in copyright infringement action; prior to trial, both parties informed court that developer's copyright infringement claims included declarations of elements of application programming interfaces (API) in allegedly infringing source code.

18. Copyrights and Intellectual Property ⚿4.5

The merger doctrine functions as an exception to the idea/expression dichotomy in copyright infringement cases; it provides that, when there are a limited number of ways to express an idea, the idea is said to "merge" with its expression, and the expression becomes unprotected.

19. Copyrights and Intellectual Property ⚿4.5

Under the merger doctrine, a court will not protect a copyrighted work from infringement if the idea contained therein can be expressed in only one way.

20. Copyrights and Intellectual Property ⚿4.5

Under the merger doctrine, when specific parts of a computer program's code, even though previously copyrighted, are the only and essential means of accom-

plishing a given task, their later use by another will not amount to infringement.

21. Copyrights and Intellectual Property ⚷10.4

Alternative expressions of idea contained in code of computer technology developer's computer program were available, and thus, idea did not merge with its expression, as would preclude copyright protection; evidence showed that developer had unlimited options as to selection and arrangement of 7000 lines of code. 17 U.S.C.A. § 101 et seq.

22. Copyrights and Intellectual Property ⚷4

Copyrightability and the scope of protectable activity are to be evaluated at the time of creation, not at the time of infringement. 17 U.S.C.A. § 102(b).

23. Copyrights and Intellectual Property ⚷10.4

Computer technology developer's declaring source code contained protectable expression that was entitled to copyright protection; portion of declaring code at issue was 7,000 lines, and developer exercised creativity in selection and arrangement of method declarations when it created packages of application programming interfaces and relevant declaring code. 17 U.S.C.A. §§ 101, 103.

24. Copyrights and Intellectual Property ⚷12(2)

The scenes a faire doctrine, which is related to the merger doctrine, operates to bar certain otherwise creative expression from copyright protection; it provides that expressive elements of a work of authorship are not entitled to protection against infringement if they are standard, stock, or common to a topic, or if they necessarily follow from a common theme or setting.

25. Copyrights and Intellectual Property ⚷12(2)

Under the scenes a faire doctrine, when certain commonplace expressions are indispensable and naturally associated with the treatment of a given idea, those expressions are treated like ideas and therefore are not protected by copyright. 17 U.S.C.A. § 102(b).

26. Copyrights and Intellectual Property ⚷12(2)

In the computer context, the scenes a faire doctrine denies copyright protection to program elements that are dictated by external factors such as the mechanical specifications of the computer on which a particular program is intended to run or widely accepted programming practices within the computer industry.

27. Copyrights and Intellectual Property ⚷12(2)

Scenes a faire doctrine did not affect copyrightability of either declaring source code in, or structure, sequence, and organization of, packages of application programming interfaces at issue in computer technology developer's infringement action against search engine operator.

28. Copyrights and Intellectual Property ⚷12(2)

In the copyright context, the focus of the scenes a faire doctrine is on the circumstances presented to the creator, not the copier.

29. Copyrights and Intellectual Property ⚷10.4

A set of commands to instruct a computer to carry out desired operations may contain expression that is eligible for copyright protection. 17 U.S.C.A. § 102(b).

30. Copyrights and Intellectual Property ⚿4.5

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. 17 U.S.C.A. § 102(b).

31. Copyrights and Intellectual Property ⚿10.4

Even if an element directs a computer to perform operations, the court must nevertheless determine whether it contains any separable expression entitled to copyright protection. 17 U.S.C.A. § 102(b).

32. Copyrights and Intellectual Property ⚿10.4

Computer technology developer's packages of application programming interfaces were not precluded from copyright protection just because they also performed functions; structure, sequence, and organization of packages was original and creative, and declaring source code could have been written and organized in any number of ways and still have achieved same functions. 17 U.S.C.A. § 102(b).

33. Copyrights and Intellectual Property ⚿10.4

To determine whether certain aspects of an allegedly infringed computer software are not protected by copyright law, the focus is on external factors that influenced the choice of the creator of the infringed product.

34. Copyrights and Intellectual Property ⚿10.4

In determining whether aspects of an allegedly infringed computer software are protected by copyright law, the focus is on the compatibility needs and programming choices of the party claiming copyright protection, not the choices the defendant made to achieve compatibility with the plaintiff's program.

35. Copyrights and Intellectual Property ⚿10.4

In determining copyrightability of computer technology developer's software, district court improperly focused its interoperability analysis on search engine operator's desires for its allegedly infringing software; whether operator's software was "interoperable" in some sense with any aspect of platform used to create software had no bearing on threshold question of whether developer's software was copyrightable. 17 U.S.C.A. § 102(b).

36. Copyrights and Intellectual Property ⚿12(1)

Copyrighted works do not lose protection when they become the industry standard.

37. Patents ⚿218(1)

Even when a patented method or system becomes an acknowledged industry standard with acquiescence of the patent owner, any permissible use generally requires payment of a reasonable royalty.

38. Copyrights and Intellectual Property ⚿53.2

The fair use doctrine both permits and requires courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster. 17 U.S.C.A. § 107.

39. Copyrights and Intellectual Property ⚿53.2

The section of the Copyright Act governing the fair use doctrine requires a case-by-case determination whether a particular use is fair. 17 U.S.C.A. § 107.

40. Copyrights and Intellectual Property ⚿88

Federal Courts ⚿3635

Fair use of a copyrighted work is a mixed question of law and fact; thus, while

subsidiary and controverted findings of fact must be reviewed for clear error, the Court of Appeals reviews the ultimate application of those facts de novo. 17 U.S.C.A. § 107.

41. Copyrights and Intellectual Property ⚿88

Where there are no material facts at issue in a copyright suit and the parties dispute only the ultimate conclusions to be drawn from those facts, the court may draw those conclusions without usurping the function of the jury.

42. Federal Courts ⚿3783

Where there are material facts in dispute and those facts have not yet been resolved by the trier of fact, appellate courts may not make findings of fact in the first instance and remand is warranted.

43. Copyrights and Intellectual Property ⚿53.2

The factor of the fair use inquiry examining the purpose and character of the use of a copyrighted work involves two sub-issues: (1) whether and to what extent the new work is transformative, and (2) whether the use serves a commercial purpose. 17 U.S.C.A. § 107.

44. Copyrights and Intellectual Property ⚿53.2

Under the fair use doctrine, the use of a copyrighted work is "transformative" if it adds something new, with a further purpose or different character, altering the first with new expression, meaning or message; the critical question is whether the new work merely supersedes the objects of the original creation or instead adds something new. 17 U.S.C.A. § 107.

See publication Words and Phrases for other judicial constructions and definitions.

45. Copyrights and Intellectual Property ⚿53.2

Under the fair use doctrine, a use is considered transformative only where a defendant changes a plaintiff's copyrighted work or uses the plaintiff's copyrighted work in a different context such that the plaintiff's work is transformed into a new creation. 17 U.S.C.A. § 107.

46. Copyrights and Intellectual Property ⚿53.2

Under the fair use doctrine, a work is not transformative where the user makes no alteration to the expressive content or message of the original copyrighted work. 17 U.S.C.A. § 107.

47. Copyrights and Intellectual Property ⚿53.2

Where the use of a copyrighted work is for the same intrinsic purpose as the copyright holder's, such use seriously weakens a claimed fair use. 17 U.S.C.A. § 107.

48. Copyrights and Intellectual Property ⚿53.2

Use of a copyrighted work that is commercial tends to weigh against a finding of fair use. 17 U.S.C.A. § 107.

49. Copyrights and Intellectual Property ⚿53.2

The more transformative a new work, the less will be the significance of other factors, like commercialism, that may weigh against a finding of fair use in a copyright infringement action. 17 U.S.C.A. § 107.

50. Copyrights and Intellectual Property ⚿12(1)

Creative expression falls within the core of a copyright's protective purposes.

51. Copyrights and Intellectual Property ⚿10.4

Because copyrighted computer programs have both functional and expressive components, where the functional components are themselves unprotected, those elements should be afforded a lower de-

gree of protection than more traditional literary works.

52. Copyrights and Intellectual Property ⚷67.3

Where the nature of a copyrighted work is such that purely functional elements exist in the work and it is necessary to copy the expressive elements in order to perform those functions, it arguably supports a finding that the use is fair. 17 U.S.C.A. § 107.

53. Copyrights and Intellectual Property ⚷53.2

Under the fair use doctrine, analysis of the factor asking a court to examine the amount and substantiality of the portion used in relation to the copyrighted work as a whole is viewed in the context of the copyrighted work, not the infringing work; the statutory language makes clear that a taking may not be excused merely because it is insubstantial with respect to the infringing work. 17 U.S.C.A. § 107(3).

54. Copyrights and Intellectual Property ⚷53.2

If a secondary user only copies as much as is necessary for his or her intended use, then the factor of the fair use doctrine asking a court to examine the amount and substantiality of the portion used in relation to the copyrighted work as a whole will not weigh against him or her. 17 U.S.C.A. § 107.

55. Copyrights and Intellectual Property ⚷53.2

Under the factor of the fair use doctrine asking a court to examine the amount and substantiality of the portion used in relation to the copyrighted work as a whole, attention turns to the persuasiveness of a parodist's justification for the particular copying done, and the enquiry will harken back to the first of the statutory factors because the extent of permissible copying varies with the purpose and character of the use. 17 U.S.C.A. § 107.

56. Copyrights and Intellectual Property ⚷53.2

The fourth and final factor of the fair use doctrine focuses on the effect of the use upon the potential market for or value of the copyrighted work; this factor reflects the idea that fair use is limited to copying by others which does not materially impair the marketability of the work which is copied. 17 U.S.C.A. § 107.

57. Copyrights and Intellectual Property ⚷53.2

The factor of the fair use doctrine focusing on the effect of the use upon the potential market for or value of the copyrighted work requires that courts consider not only the extent of market harm caused by the particular actions of the alleged infringer, but also whether unrestricted and widespread conduct of the sort engaged in by the defendant would result in a substantially adverse impact on the potential market for the original. 17 U.S.C.A. § 107.

58. Copyrights and Intellectual Property ⚷53.2

Market harm is a matter of degree, and the importance of the factor of the fair use doctrine focusing on the effect of the use upon the potential market for or value of the copyrighted work will vary, not only with the amount of harm, but also with the relative strength of the showing on the other factors. 17 U.S.C.A. § 107.

59. Copyrights and Intellectual Property ⚷89(2)

Genuine dispute of material fact existed as to whether search engine operator's use of computer technology developer's copyrighted software was transformative, precluding summary judgment on operator's fair use defense to copyright infringement claims. 17 U.S.C.A. § 107.

60. Courts ⚿96(7)

When reviewing a district court's grant or denial of a motion for judgment as a matter of law (JMOL), the Court of Appeals applies the procedural law of the relevant regional circuit.

61. Federal Courts ⚿3605

The Court of Appeals reviews a district court's judgment as a matter of law (JMOL) decision de novo, applying the same standard as the district court.

62. Federal Civil Procedure ⚿2152, 2608.1

To grant judgment as a matter of law (JMOL), a court must find that the evidence presented at trial permits only one reasonable conclusion and that no reasonable juror could find in the non-moving party's favor.

63. Copyrights and Intellectual Property ⚿67.3

Even assuming that Ninth Circuit recognized stand-alone de minimis defense to copyright infringement, search engine operator's copying of computer technology developer's software was not de minimus; although operator argued that eight decompiled files were insignificant because they were used only to test platform adapted for mobile devices, developer's expert testified that "using the copied files even as test files would have been significant use" and district court specifically found that there was no testimony to the contrary.

E. Joshua Rosenkranz, Orrick, Herrington & Sutcliffe LLP, of New York, New York, argued for plaintiff-appellant. With him on the brief were Mark S. Davies, Andrew D. Silverman, Kelly M. Daley; and Annette L. Hurst, Gabriel M. Ramsey, and Elizabeth C. McBride, of San Francisco, CA. Of counsel on the brief were Dorian E. Daley, Deborah K. Miller, Matthew Sarboraria, and Andrew C. Temkin, Oracle America, Inc., of Redwood Shores, CA; and Dale M. Cendali, Diana M. Torres, Sean B. Fernandes, and Joshua L. Simmons, Kirkland & Ellis LLP, of New York, NY. Of counsel were Susan M. Davies, Kirkland & Ellis LLP, of New York, NY; Michael A. Jacobs, Morrison & Foerster LLP, of San Francisco, CA; and Kenneth A. Kuwayti, of Palo Alto, CA.

Robert A. Van Nest, Keker & Van Nest LLP, of San Francisco, CA, argued for defendant-cross-appellant. With him on the brief were Christa M. Anderson, Steven A. Hirsch, Michael S. Kwun, and Daniel E. Jackson. Of counsel on the brief were Ian C. Ballon and Heather Meeker, Greenberg Traurig, LLP, of East Palo Alto, CA; Renny Hwang, Google Inc., of Mountain View, CA; and Daryl L. Joseffer and Bruce W. Baber, King & Spalding LLP, of Washington, DC.

Marcia B. Paul, Davis Wright Tremaine LLP, of New York, NY, for amicus curiae Ralph Oman. With her on the brief were Lacy H. Koonce, III and Deborah A. Adler.

William A. Rudy, Lathrop & Gage LLP, of Kansas City, MO, for amici curiae Picture Archive Council of America, Inc., et al. With him on the brief were Carole E. Handler and Brianna E. Dahlberg, of Los Angeles, CA.

Gregory G. Garre, Latham & Watkins, LLP, of Washington, DC, for amici curiae Microsoft Corporation, et al. With him on the brief was Lori Alvino McGill. Of counsel on the brief were Paul T. Dacier, Krishnendu Gupta, EMC Corporation, of Hopkinton, MA; and Douglas Luftman, Netapp, Inc., of Sunnyvale, CA.

Jared Bobrow, Weil, Gotshal & Manges LLP, of Redwood Shores, CA, for amici curiae Eugene H. Spafford, Ph.D., et al.

With him on the brief was Aaron Y. Huang.

Matthew S. Hellman, Jenner & Block LLP, of Washington, DC, for amicus curiae BSA/The Software Alliance. With him on the brief was Paul M. Smith.

Steven T. Cottreau, Clifford Chance U.S. LLP, of Washington, DC, for amici curiae, Scott McNealy, et al.

Meredith Jacob, Program on Information Justice and Intellectual Property, American University, Washington College of Law, of Washington, DC, for amici curiae Intellectual Property Law Professors.

Julie P. Samuels, Electronic Frontier Foundation, of San Francisco, CA, for amici curiae Computer Scientists. With her on the brief was Michael Barclay. Of counsel on the brief was Jason M. Schultz, NYU Technology Law and Policy Clinic, NYU School of Law, of New York, NY.

Jonathan Band, Jonathan Band PLLC, of Washington, DC, filed a brief for amicus curiae Computer & Communications Industry Association. With him on the brief was Matthew Schruers, Computer & Communications Industry Association, of Washington, DC.

Chad Ruback, The Ruback Law Firm, of Dallas, TX, filed a brief for amici curiae Rackspace US, Inc., et al.

Jennifer M. Urban, Samuelson Law, Technology and Public Policy Clinic, U.C. Berkeley School of Law, of Berkeley, CA for amici curiae Software Innovators, et al.

Before O'MALLEY, PLAGER, and TARANTO, Circuit Judges.

O'MALLEY, Circuit Judge.

This copyright dispute involves 37 packages of computer source code. The parties have often referred to these groups of computer programs, individually or collectively, as "application programming interfaces," or API packages, but it is their content, not their name, that matters. The predecessor of Oracle America, Inc. ("Oracle") wrote these and other API packages in the Java programming language, and Oracle licenses them on various terms for others to use. Many software developers use the Java language, as well as Oracle's API packages, to write applications (commonly referred to as "apps") for desktop and laptop computers, tablets, smartphones, and other devices.

Oracle filed suit against Google Inc. ("Google") in the United States District Court for the Northern District of California, alleging that Google's Android mobile operating system infringed Oracle's patents and copyrights. The jury found no patent infringement, and the patent claims are not at issue in this appeal. As to the copyright claims, the parties agreed that the jury would decide infringement, fair use, and whether any copying was de minimis, while the district judge would decide copyrightability and Google's equitable defenses. The jury found that Google infringed Oracle's copyrights in the 37 Java packages and a specific computer routine called "rangeCheck," but returned a noninfringement verdict as to eight decompiled security files. The jury deadlocked on Google's fair use defense.

After the jury verdict, the district court denied Oracle's motion for judgment as a matter of law ("JMOL") regarding fair use as well as Google's motion for JMOL with respect to the rangeCheck files. Order on Motions for Judgment as a Matter of Law, *Oracle Am., Inc. v. Google Inc.,* No. 3:10–cv–3561 (N.D.Cal. May 10, 2012), ECF No. 1119. Oracle also moved for JMOL of infringement with respect to the eight decompiled security files. In granting that motion, the court found that: (1) Google admitted to copying the eight files; and (2) no reasonable jury could find that the copying was de minimis. *Oracle Am., Inc. v. Google Inc.,* No. C 10–3561, 2012 U.S.

Dist. LEXIS 66417 (N.D.Cal. May 11, 2012) ("*Order Granting JMOL on Decompiled Files*").

Shortly thereafter, the district court issued its decision on copyrightability, finding that the replicated elements of the 37 API packages—including the declaring code and the structure, sequence, and organization—were not subject to copyright protection. *Oracle Am., Inc. v. Google Inc.*, 872 F.Supp.2d 974 (N.D.Cal.2012) ("*Copyrightability Decision*"). Accordingly, the district court entered final judgment in favor of Google on Oracle's copyright infringement claims, except with respect to the rangeCheck code and the eight decompiled files. Final Judgment, *Oracle Am., Inc. v. Google Inc.*, No. 3:10–cv3561, 2012 WL 9028839 (N.D.Cal. June 20, 2012), ECF No. 1211. Oracle appeals from the portion of the final judgment entered against it, and Google cross-appeals from the portion of that same judgment entered in favor of Oracle as to the rangeCheck code and eight decompiled files.

Because we conclude that the declaring code and the structure, sequence, and organization of the API packages are entitled to copyright protection, we reverse the district court's copyrightability determination with instructions to reinstate the jury's infringement finding as to the 37 Java packages. Because the jury deadlocked on fair use, we remand for further consideration of Google's fair use defense in light of this decision. With respect to Google's cross-appeal, we affirm the district court's decisions: (1) granting Oracle's motion for JMOL as to the eight decompiled Java files that Google copied into Android; and (2) denying Google's motion for JMOL with respect to the rangeCheck function. Accordingly, we affirm-in-part, reverse-in-part, and remand for further proceedings.

Background

A. The Technology

Sun Microsystems, Inc. ("Sun") developed the Java "platform" for computer programming and released it in 1996.[1] The aim was to relieve programmers from the burden of writing different versions of their computer programs for different operating systems or devices. "The Java platform, through the use of a virtual machine, enable[d] software developers to write programs that [we]re able to run on different types of computer hardware without having to rewrite them for each different type." *Copyrightability Decision*, 872 F.Supp.2d at 977. With Java, a software programmer could "write once, run anywhere."

The Java virtual machine ("JVM") plays a central role in the overall Java platform. The Java programming language itself—which includes words, symbols, and other units, together with syntax rules for using them to create instructions—is the language in which a Java programmer writes source code, the version of a program that is "in a human-readable language." *Id.* For the instructions to be executed, they must be converted (or compiled) into binary machine code (object code) consisting of 0s and 1s understandable by the particular computing device. In the Java system, "source code is first converted into 'bytecode,' an intermediate form, before it is then converted into binary machine code by the Java virtual machine" that has been designed for that device. *Id.* The Java platform includes the "Java development kit (JDK), javac compiler, tools and utilities, runtime programs, class libraries (API packages), and the Java virtual machine." *Id.* at 977 n. 2.

Sun wrote a number of ready-to-use Java programs to perform common com-

1. Oracle acquired Sun in 2010.

puter functions and organized those programs into groups it called "packages." These packages, which are the application programming interfaces at issue in this appeal, allow programmers to use the prewritten code to build certain functions into their own programs, rather than write their own code to perform those functions from scratch. They are shortcuts. Sun called the code for a specific operation (function) a "method." It defined "classes" so that each class consists of specified methods plus variables and other elements on which the methods operate. To organize the classes for users, then, it grouped classes (along with certain related "interfaces") into "packages." *See id.* at 982 (describing organization: "[e]ach package [i]s broken into classes and those in turn [are] broken into methods"). The parties have not disputed the district court's analogy: Oracle's collection of API packages is like a library, each package is like a bookshelf in the library, each class is like a book on the shelf, and each method is like a how-to chapter in a book. *Id.* at 977.

The original Java Standard Edition Platform ("Java SE") included "eight packages of pre-written programs." *Id.* at 982. The district court found, and Oracle concedes to some extent, that three of those packages—java.lang,java.io, and java.util—were "core" packages, meaning that programmers using the Java language had to use them "in order to make any worthwhile use of the language." *Id.* By 2008, the Java platform had more than 6,000 methods making up more than 600 classes grouped into 166 API packages. There are 37 Java API packages at issue in this appeal, three of which are the core packages identified by the district court.[2] These packages contain thousands of individual elements, including classes, subclasses, methods, and interfaces.

Every package consists of two types of source code—what the parties call (1) declaring code; and (2) implementing code. Declaring code is the expression that identifies the prewritten function and is sometimes referred to as the "declaration" or "header." As the district court explained, the "main point is that this header line of code introduces the method body and specifies very precisely the inputs, name and other functionality." *Id.* at 979–80. The expressions used by the programmer from the declaring code command the computer to execute the associated implementing code, which gives the computer the step-by-step instructions for carrying out the declared function.

To use the district court's example, one of the Java API packages at issue is "java.lang." Within that package is a class called "math," and within "math" there are several methods, including one that is designed to find the larger of two numbers: "max." The declaration for the "max" method, as defined for integers, is: "public static int max(int x, int y)," where the word "public" means that the method is generally accessible, "static" means that no specific instance of the class is needed to call the method, the first "int" indicates that the method returns an integer, and "int x" and "int y" are the two numbers (inputs) being compared. *Copyrightability*

2. The 37 API packages involved in this appeal are: java.awt.font, java.beans, java.io, java.lang, java. lang. annotation, java.lang.ref, java. lang. reflect,java.net, java.nio, java. nio. channels, java.nio.channels.spi, java.nio.charset, java.nio.charset.spi, java.security, java. security.acl, java. security. cert, java. security. interfaces, java. security. spec, java.sql, java. text, java.util, java.util.jar, java.util.logging, java.util.prefs, java.util.regex, java.util.zip, javax.crypto, javax. crypto. interfaces, javax.crypto.spec,javax.net, javax.net.ssl, javax. security. auth, javax. security.auth.callback, javax. security. auth.login, javax. security. auth.x500, javax.security.cert, and javax.sql.

Decision, 872 F.Supp.2d at 980–82. A programmer calls the "max" method by typing the name of the method stated in the declaring code and providing unique inputs for the variables "x" and "y." The expressions used command the computer to execute the implementing code that carries out the operation of returning the larger number.

Although Oracle owns the copyright on Java SE and the API packages, it offers three different licenses to those who want to make use of them. The first is the General Public License, which is free of charge and provides that the licensee can use the packages—both the declaring and implementing code—but must "contribute back" its innovations to the public. This arrangement is referred to as an "open source" license. The second option is the Specification License, which provides that the licensee can use the declaring code and organization of Oracle's API packages but must write its own implementing code. The third option is the Commercial License, which is for businesses that "want to use and customize the full Java code in their commercial products and keep their code secret." Appellant Br. 14. Oracle offers the Commercial License in exchange for royalties. To maintain Java's "write once, run anywhere" motto, the Specification and Commercial Licenses require that the licensees' programs pass certain tests to ensure compatibility with the Java platform.

The testimony at trial also revealed that Sun was licensing a derivative version of the Java platform for use on mobile devices: the Java Micro Edition ("Java ME"). Oracle licensed Java ME for use on feature phones and smartphones. Sun/Oracle has never successfully developed its own smartphone platform using Java.

B. Google's Accused Product: Android

The accused product is Android, a software platform that was designed for mobile devices and competes with Java in that market. Google acquired Android, Inc. in 2005 as part of a plan to develop a smartphone platform. Later that same year, Google and Sun began discussing the possibility of Google "taking a license to use and to adapt the entire Java platform for mobile devices." *Copyrightability Decision*, 872 F.Supp.2d at 978. They also discussed a "possible co-development partnership deal with Sun under which Java technology would become an open-source part of the Android platform, adapted for mobile devices." *Id.* The parties negotiated for months but were unable to reach an agreement. The point of contention between the parties was Google's refusal to make the implementation of its programs compatible with the Java virtual machine or interoperable with other Java programs. Because Sun/Oracle found that position to be anathema to the "write once, run anywhere" philosophy, it did not grant Google a license to use the Java API packages.

When the parties' negotiations reached an impasse, Google decided to use the Java programming language to design its own virtual machine—the Dalvik virtual machine ("Dalvik VM")—and "to write its own implementations for the functions in the Java API that were key to mobile devices." *Id.* Google developed the Android platform, which grew to include 168 API packages—37 of which correspond to the Java API packages at issue in this appeal.

With respect to the 37 packages at issue, "Google believed Java application programmers would want to find the same 37 sets of functionalities in the new Android system callable by the same names as used in Java." *Id.* To achieve this result, Google copied the declaring source code from the

37 Java API packages verbatim, inserting that code into parts of its Android software. In doing so, Google copied the elaborately organized taxonomy of all the names of methods, classes, interfaces, and packages—the "overall system of organized names—covering 37 packages, with over six hundred classes, with over six thousand methods." *Copyrightability Decision*, 872 F.Supp.2d at 999. The parties and district court referred to this taxonomy of expressions as the "structure, sequence, and organization" or "SSO" of the 37 packages. It is undisputed, however, that Google wrote its own implementing code, except with respect to: (1) the rangeCheck function, which consisted of nine lines of code; and (2) eight decompiled security files.

As to rangeCheck, the court found that the Sun engineer who wrote it later worked for Google and contributed two files he created containing the rangeCheck function—"Timsort.java" and "ComparableTimsort"—to the Android platform. In doing so, the nine-line rangeCheck function was copied directly into Android. As to the eight decompiled files, the district court found that they were copied and used as test files but "never found their way into Android or any handset." *Id.* at 983.

Google released the Android platform in 2007, and the first Android phones went on sale the following year. Although it is undisputed that certain Android software contains copies of the 37 API packages' declaring code at issue, neither the district court nor the parties specify in which programs those copies appear. Oracle indicated at oral argument, however, that all Android phones contain copies of the accused portions of the Android software. Oral Argument at 1:35, *available at* http://www.cafc.uscourts.gov/oral–argument–recordings/2013–1021/all. Android smartphones "rapidly grew in popularity and now comprise a large share of the United States market." *Copyrightability Decision*, 872 F.Supp.2d at 978. Google provides the Android platform free of charge to smartphone manufacturers and receives revenue when customers use particular functions on the Android phone. Although Android uses the Java programming language, it is undisputed that Android is not generally Java compatible. As Oracle explains, "Google ultimately designed Android to be *incompatible* with the Java platform, so that apps written for one will not work on the other." Appellant Br. 29.

C. Trial and Post–Trial Rulings

Beginning on April 16, 2012, the district court and the jury—on parallel tracks-viewed documents and heard testimony from twenty—four witnesses on copyrightability, infringement, fair use, and Google's other defenses. Because the parties agreed the district court would decide copyrightability, the court instructed the jury to assume that the structure, sequence, and organization of the 37 API packages was copyrightable. And, the court informed the jury that Google conceded that it copied the declaring code used in the 37 packages verbatim. The court also instructed the jury that Google conceded copying the rangeCheck function and the eight decompiled security files, but that Google maintained that its use of those lines of code was de minimis. *See* Final Charge to the Jury (Phase One), *Oracle Am., Inc. v. Google Inc.*, 3:10–cv–3561 (N.D.Cal. Apr. 30, 2012), ECF No. 1018 at 14 ("With respect to the infringement issues concerning the rangeCheck and other similar files, Google agrees that the accused lines of code and comments came from the copyrighted material but contends that the amounts involved were so negligible as to be de minimis and thus should be excused.").

On May 7, 2012, the jury returned a verdict finding that Google infringed Ora-

cle's copyright in the 37 Java API packages and in the nine lines of rangeCheck code, but returned a noninfringement verdict as to eight decompiled security files. The jury hung on Google's fair use defense.

The parties filed a number of post-trial motions, most of which were ultimately denied. In relevant part, the district court denied Oracle's motion for JMOL regarding fair use and Google's motion for JMOL as to the rangeCheck files. Order on Motions for Judgment as a Matter of Law, *Oracle Am., Inc. v. Google Inc.,* No. 3:10–cv–3561 (N.D.Cal. May 10, 2012), ECF No. 1119. The district court granted Oracle's motion for JMOL of infringement as to the eight decompiled files, however. In its order, the court explained that: (1) Google copied the files in their entirety; (2) the trial testimony revealed that the use of those files was "significant"; and (3) no reasonable jury could find the copying de minimis. *Order Granting JMOL on Decompiled Files,* 2012 U.S. Dist. LEXIS 66417, at *6.

On May 31, 2012, the district court issued the primary decision at issue in this appeal, finding that the replicated elements of the Java API packages—including the declarations and their structure, sequence, and organization—were not copyrightable. As to the declaring code, the court concluded that "there is only one way to write" it, and thus the "merger doctrine bars anyone from claiming exclusive copyright ownership of that expression." *Copyrightability Decision,* 872 F.Supp.2d at 998. The court further found that the declaring code was not protectable because "names and short phrases cannot be copyrighted." *Id.* As such, the court determined that "there can be no copyright violation in using the identical declarations." *Id.*

As to the overall structure, sequence, and organization of the Java API packages, the court recognized that "nothing in the rules of the Java language ... required that Google replicate the same groupings even if Google was free to replicate the same functionality." *Id.* at 999. Therefore, the court determined that "Oracle's best argument ... is that while no single name is copyrightable, Java's overall system of organized names—covering 37 packages, with over six hundred classes, with over six thousand methods—is a 'taxonomy' and, therefore, copyrightable." *Id.*

Although it acknowledged that the overall structure of Oracle's API packages is creative, original, and "resembles a taxonomy," the district court found that it "is nevertheless a command structure, a system or method of operation—a long hierarchy of over six thousand commands to carry out pre-assigned functions"—that is not entitled to copyright protection under Section 102(b) of the Copyright Act. *Id.* at 999–1000. In reaching this conclusion, the court emphasized that, "[o]f the 166 Java packages, 129 were not violated in any way." *Id.* at 1001. And, of the 37 Java API packages at issue, "97 percent of the Android lines were new from Google and the remaining three percent were freely replicable under the merger and names doctrines." *Id.* On these grounds, the court dismissed Oracle's copyright claims, concluding that "the particular elements replicated by Google were free for all to use under the Copyright Act." *Id.*

On June 20, 2012, the district court entered final judgment in favor of Google and against Oracle on its claim for copyright infringement, except with respect to the rangeCheck function and the eight decompiled files. As to rangeCheck and the decompiled files, the court entered judgment for Oracle and against Google in the amount of zero dollars, per the parties' stipulation. Final Judgment, *Oracle Am., Inc. v. Google Inc.,* No. 3:10–cv3561

(N.D.Cal. June 20, 2012), ECF No. 1211. Oracle timely appealed from the portion of the district court's final judgment entered against it and Google timely cross-appealed with respect to rangeCheck and the eight decompiled files. Because this action included patent claims, we have jurisdiction pursuant to 28 U.S.C. § 1295(a)(1).

Discussion

I. Oracle's Appeal

It is undisputed that the Java programming language is open and free for anyone to use. Except to the limited extent noted below regarding three of the API packages, it is also undisputed that Google could have written its own API packages using the Java language. Google chose not to do that. Instead, it is undisputed that Google copied 7,000 lines of declaring code and generally replicated the overall structure, sequence, and organization of Oracle's 37 Java API packages. The central question before us is whether these elements of the Java platform are entitled to copyright protection. The district court concluded that they are not, and Oracle challenges that determination on appeal. Oracle also argues that the district court should have dismissed Google's fair use defense as a matter of law.

According to Google, however, the district court correctly determined that: (1) there was only one way to write the Java method declarations and remain "interoperable" with Java; and (2) the organization and structure of the 37 Java API packages is a "command structure" excluded from copyright protection under Section 102(b). Google also argues that, if we reverse the district court's copyrightability determination, we should direct the district court to retry its fair use defense.

[1–3] "When the questions on appeal involve law and precedent on subjects not exclusively assigned to the Federal Circuit, the court applies the law which would be applied by the regional circuit." *Atari Games Corp. v. Nintendo of Am., Inc.*, 897 F.2d 1572, 1575 (Fed.Cir.1990). Copyright issues are not exclusively assigned to the Federal Circuit. *See* 28 U.S.C. § 1295. The parties agree that Ninth Circuit law applies and that, in the Ninth Circuit, whether particular expression is protected by copyright law is "subject to de novo review." *Ets–Hokin v. Skyy Spirits, Inc.*, 225 F.3d 1068, 1073 (9th Cir.2000).[3]

3. The Supreme Court has not addressed whether copyrightability is a pure question of law or a mixed question of law and fact, or whether, if it is a mixed question of law and fact, the factual components of that inquiry are for the court, rather than the jury. Relatedly, it has not decided the standard of review that applies on appeal. Ten years ago, before finding it unnecessary to decide whether copyrightability is a pure question of law or a mixed question of law and fact, the Seventh Circuit noted that it had "found only a handful of appellate cases addressing the issue, and they are split." *Gaiman v. McFarlane*, 360 F.3d 644, 648 (7th Cir.2004). And, panels of the Ninth Circuit have defined the respective roles of the jury and the court differently where questions of originality were at issue. *Compare North Coast Indus. v. Jason Maxwell, Inc.*, 972 F.2d 1031, 1035 (9th Cir. 1992), *with Ets–Hokin*, 225 F.3d at 1073. More recently, several district courts within the Ninth Circuit have treated copyrightability as a question for only the court, regardless of whether it is a pure question of law. *See Stern v. Does*, No. 09–1986, 2011 U.S. Dist. LEXIS 37735, *7 (C.D.Cal. Feb. 10, 2011); *Jonathan Browning, Inc. v. Venetian Casino Resort LLC*, No. C 07–3983, 2009 U.S. Dist. LEXIS 57525, at *2 (N.D.Cal. June 19, 2009); *see also Pivot Point Int'l, Inc. v. Charlene Prods., Inc.*, 932 F.Supp. 220, 225 (N.D.Ill. 1996) (Easterbrook, J.) (citing to *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996), and concluding that whether works are copyrightable is a question which the "jury has nothing to do with"). We need not address any of these questions, because the parties here agreed that the district court would decide copyright-

We are mindful that the application of copyright law in the computer context is often a difficult task. *See Lotus Dev. Corp. v. Borland Int'l, Inc.,* 49 F.3d 807, 820 (1st Cir.1995) (Boudin, J., concurring) ("Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit."). On this record, however, we find that the 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. The court also erred by importing fair use principles, including interoperability concerns, into its copyrightability analysis.

For the reasons that follow, we conclude that the declaring code and the structure, sequence, and organization of the 37 Java API packages are entitled to copyright protection. Because there is an insufficient record as to the relevant fair use factors, we remand for further proceedings on Google's fair use defense.

A. Copyrightability

[4] The Copyright Act provides protection to "original works of authorship fixed in any tangible medium of expression," including "literary works." 17 U.S.C. § 102(a). It is undisputed that computer programs—defined in the Copyright Act as "a 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—can be subject to copyright protection as "literary works." *See Atari Games Corp. v. Nintendo of Am., Inc.,* 975 F.2d 832, 838 (Fed.Cir.1992) ("As literary works, copyright protection extends to computer programs."). Indeed, the legislative history explains that "literary works" includes "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. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667.

[5, 6] By statute, a work must be "original" to qualify for copyright protection. 17 U.S.C. § 102(a). This "originality requirement is not particularly stringent," however. *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.,* 499 U.S. 340, 358, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991). "Original, as the term is used in copyright, 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." *Id.* at 345, 111 S.Ct. 1282.

[7] Copyright protection extends only to the expression of an idea—not to the underlying idea itself. *Mazer v. Stein,* 347 U.S. 201, 217, 74 S.Ct. 460, 98 L.Ed. 630 (1954) ("Unlike a patent, a copyright gives no exclusive right to the art disclosed; protection is given only to the expression of the idea—not the idea itself."). This distinction—commonly referred to as the "idea/expression dichotomy"—is codified in Section 102(b) of the Copyright Act, which provides:

> 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.

17 U.S.C. § 102(b); *see Golan v. Holder,* —— U.S. ——, 132 S.Ct. 873, 890, 181 L.Ed.2d 835 (2012) ("The idea/expression

ability, and both largely agree that we may undertake a review of that determination de novo.

dichotomy is codified at 17 U.S.C. § 102(b).").

The idea/expression dichotomy traces back to the Supreme Court's decision in *Baker v. Selden*, 101 U.S. 99, 101, 11 Otto 99, 25 L.Ed. 841 (1879). In *Baker*, the plaintiff Selden wrote and obtained copyrights on a series of books setting out a new system of bookkeeping. *Id.* at 100. The books included an introductory essay explaining the system and blank forms with ruled lines and headings designed for use with that system. *Id.* Baker published account books employing a system with similar forms, and Selden filed suit alleging copyright infringement. According to Selden, the "ruled lines and headings, given to illustrate the system, are a part of the book" and "no one can make or use similar ruled lines and headings, or ruled lines and headings made and arranged on substantially the same system, without violating the copyright." *Id.* at 101.

The Supreme Court framed the issue on appeal in *Baker* as "whether the exclusive property in a system of book-keeping can be claimed, under the law of copyright, by means of a book in which that system is explained." *Id.* In reversing the circuit court's decision, the Court concluded that the "copyright of a book on book-keeping cannot secure the exclusive right to make, sell, and use account-books prepared upon the plan set forth in such book." *Id.* at 104. Likewise, the "copyright of a work on mathematical science cannot give to the author an exclusive right to the methods of operation which he propounds." *Id.* at 103. The Court found that, although the copyright protects the way Selden "explained and described a peculiar system of book-keeping," it does not prevent others from using the system described therein. *Id.* at 104. The Court further 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 id.* (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").

Courts routinely cite *Baker* as the source of several principles incorporated into Section 102(b) that relate to this appeal, including that: (1) copyright protection extends only to expression, not to ideas, systems, or processes; and (2) "those elements of a computer program that are necessarily incidental to its function are . . . unprotectable." *See Computer Assocs. Int'l v. Altai, Inc.*, 982 F.2d 693, 704–05 (2d Cir.1992) ("*Altai* ") (discussing *Baker*, 101 U.S. at 103–04).

[8] It is well established that copyright protection can extend to both literal and non-literal elements of a computer program. *See Altai*, 982 F.2d at 702. The literal elements of a computer program are the source code and object code. *See Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989). Courts have defined source code as "the spelled-out program commands that humans can read." *Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 533 (6th Cir.2004). Object code refers to "the binary language comprised of zeros and ones through which the computer directly receives its instructions." *Altai*, 982 F.2d at 698. Both source and object code "are consistently held protected by a copyright on the program." *Johnson Controls*, 886 F.2d at 1175; *see also Altai*, 982 F.2d at 702 ("It is now well settled that the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection."). Google nowhere disputes that premise. *See, e.g.*, Oral Argument at 57:38.

[9] The non-literal components of a computer program include, among other things, the program's sequence, structure,

and organization, as well as the program's user interface. *Johnson Controls,* 886 F.2d at 1175. As discussed below, whether the non-literal elements of a program "are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself." *Id.*

In this case, Oracle claims copyright protection with respect to both: (1) literal elements of its API packages—the 7,000 lines of declaring source code; and (2) non-literal elements—the structure, sequence, and organization of each of the 37 Java API packages.

[10] The distinction between literal and non-literal aspects of a computer program is separate from the distinction between literal and non-literal copying. *See Altai,* 982 F.2d at 701–02. "Literal" copying is verbatim copying of original expression. "Non-literal" copying is "paraphrased or loosely paraphrased rather than word for word." *Lotus Dev. Corp. v. Borland Int'l,* 49 F.3d 807, 814 (1st Cir. 1995). Here, Google concedes that it copied the declaring code verbatim. Oracle explains that the lines of declaring code "embody the structure of each [API] package, just as the chapter titles and topic sentences represent the structure of a novel." Appellant Br. 45. As Oracle explains, when Google copied the declaring code in these packages "it also copied the 'sequence and organization' of the packages (i.e., the three-dimensional structure with all the chutes and ladders)" employed by Sun/Oracle in the packages. Appellant Br. 27. Oracle also argues that the nonliteral elements of the API packages—the structure, sequence, and organization that led naturally to the implementing code Google created—are entitled to protection. Oracle does not assert "literal" copying of the entire SSO, but, rather, that Google literally copied the declaring code and then paraphrased the remainder of the SSO by writing its own implementing code. It therefore asserts non-literal copying with respect to the entirety of the SSO.

At this stage, it is undisputed that the declaring code and the structure and organization of the Java API packages are original. 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. 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. *See Copyrightability Decision,* 872 F.Supp.2d at 976 ("The overall name tree, of course, has creative elements . . . ."); *Id.* at 999 ("Yes, it is creative. Yes, it is original."); Appellee Br. 5 ("Google does not dispute" the district court's finding that "the Java API clears the low originality threshold."). The court found, however, that neither the declaring code nor the SSO was entitled to copyright protection under the Copyright Act.

[11] Although the parties agree that Oracle's API packages meet the originality requirement under Section 102(a), they disagree as to the proper interpretation and application of Section 102(b). For its part, Google suggests that there is a two-step copyrightability analysis, wherein Section 102(a) grants copyright protection to original works, while Section 102(b) takes it away if the work has a functional component. To the contrary, however, Congress emphasized that Section 102(b) "in no way enlarges or contracts the scope of copyright protection" and that its "purpose is to restate . . . that the basic dichotomy between expression and idea remains unchanged." *Feist,* 499 U.S. at 356, 111 S.Ct. 1282 (quoting H.R.Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670). "Section 102(b)

does not extinguish the protection accorded a particular expression of an idea merely because that expression is embodied in a method of operation." *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1372 (10th Cir.1997). Section 102(a) and 102(b) are to be considered collectively so that certain expressions are subject to greater scrutiny. *Id.* In assessing copyrightability, the district court is required to ferret out apparent expressive aspects of a work and then separate protectable expression from "unprotectable ideas, facts, processes, and methods of operation." *See Atari*, 975 F.2d at 839.

Of course, as with many things, in defining this task, the devil is in the details. Circuit courts have struggled with, and disagree over, the tests to be employed when attempting to draw the line between what is protectable expression and what is not. *Compare Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir.1986) (everything not necessary to the purpose or function of a work is expression), *with Lotus*, 49 F.3d at 815 (methods of operation are means by which a user operates something and any words used to effectuate that operation are unprotected expression). When assessing whether the non-literal elements of a computer program constitute protectable expression, the Ninth Circuit has endorsed an "abstraction-filtration-comparison" test formulated by the Second Circuit and expressly adopted by several other circuits. *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1525 (9th Cir.1992) ("In our view, in light of the essentially utilitarian nature of computer programs, the Second Circuit's approach is an appropriate one."). This test rejects the notion that anything that performs a function is necessarily uncopyrightable. *See Mitel*, 124 F.3d at 1372 (rejecting the *Lotus* court's formulation, and concluding that, "although an element of a work may be characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection."). And it also rejects as flawed the *Whelan* assumption that, once any separable idea can be identified in a computer program everything else must be protectable expression, on grounds that more than one idea may be embodied in any particular program. *Altai*, 982 F.2d at 705–06.

[12–14] Thus, this test eschews bright line approaches and requires a more nuanced assessment of the particular program at issue in order to determine what expression is protectable and infringed. As the Second Circuit explains, this test has three steps. In the abstraction step, the court "first break[s] down the allegedly infringed program into its constituent structural parts." *Id.* at 706. In the filtration step, the court "sift[s] out all nonprotectable material," including ideas and "expression that is necessarily incidental to those ideas." *Id.* In the final step, the court compares the remaining creative expression with the allegedly infringing program.[4]

[15] In the second step, the court is first to assess whether the expression is original to the programmer or author. *Atari*, 975 F.2d at 839. The court must then determine whether the particular inclusion of any level of abstraction is dictated by considerations of efficiency, required by factors already external to the program

4. Importantly, this full analysis only applies where a copyright owner alleges infringement of the non-literal aspects of its work. Where "admitted literal copying of a discrete, easily-conceptualized portion of a work" is at issue—as with Oracle's declaring code—a court "need not perform a complete abstraction-filtration-comparison analysis" and may focus the protectability analysis on the filtration stage, with attendant reference to standard copyright principles. *Mitel*, 124 F.3d at 1372–73.

itself, or taken from the public domain—all of which would render the expression unprotectable. *Id.* These conclusions are to be informed by traditional copyright principles of originality, merger, and scenes a faire. *See Mitel,* 124 F.3d at 1372 ("Although this core of expression is eligible for copyright protection, it is subject to the rigors of filtration analysis which excludes from protection expression that is in the public domain, otherwise unoriginal, or subject to the doctrines of merger and scenes a faire.").

In all circuits, it is clear that the first step is part of the copyrightability analysis and that the third is an infringement question. It is at the second step of this analysis where the circuits are in less accord. Some treat all aspects of this second step as part of the copyrightability analysis, while others divide questions of originality from the other inquiries, treating the former as a question of copyrightability and the latter as part of the infringement inquiry. *Compare Lexmark,* 387 F.3d at 537–38 (finding that the district court erred in assessing principles of merger and scenes a faire in the infringement analysis, rather than as a component of copyrightability), *with Kregos,* 937 F.2d at 705 (noting that the Second Circuit has considered the merger doctrine "in determining whether actionable infringement has occurred, rather than whether a copyright is valid"); *see also Lexmark,* 387 F.3d at 557 (Feikens, J., dissenting-in-part) (noting the circuit split and concluding that, where a court is assessing merger of an expression with a method of operation, "I would find the merger doctrine can operate only as a defense to infringement in that context, and as such has no bearing on the question of copyrightability."). We need not assess the wisdom of these respective views because there is no doubt on which side of this circuit split the Ninth Circuit falls.

[16] In the Ninth Circuit, while questions regarding originality are considered questions of copyrightability, concepts of merger and scenes a faire are affirmative defenses to claims of infringement. *Ets–Hokin,* 225 F.3d at 1082; *Satava v. Lowry,* 323 F.3d 805, 810 n. 3 (9th Cir.2003) ("The Ninth Circuit treats scenes a faire as a defense to infringement rather than as a barrier to copyrightability."). The Ninth Circuit has acknowledged that "there is some disagreement among courts as to whether these two doctrines figure into the issue of copyrightability or are more properly defenses to infringement." *Ets–Hokin,* 225 F.3d at 1082 (citations omitted). It, nonetheless, has made clear that, in that circuit, these concepts are to be treated as defenses to infringement. *Id.* (citing *Kregos,* 937 F.2d at 705 (holding that the merger doctrine relates to infringement, not copyrightability); *Reed–Union Corp. v. Turtle Wax, Inc.,* 77 F.3d 909, 914 (7th Cir.1996) (explaining why the doctrine of scenes a faire is separate from the validity of a copyright)).

With these principles in mind, we turn to the trial court's analysis and judgment and to Oracle's objections thereto. While the trial court mentioned the abstraction-filtration-comparison test when describing the development of relevant law, it did not purport to actually apply that test. Instead, it moved directly to application of familiar principles of copyright law when assessing the copyrightability of the declaring code and interpreted Section 102(b) to preclude copyrightability for any functional element "essential for interoperability" "regardless of its form." *Copyrightability Decision,* 872 F.Supp.2d at 997.

Oracle asserts that all of the trial court's conclusions regarding copyrightability are erroneous. Oracle argues that its Java API packages are entitled to protection under the Copyright Act because they are

expressive and could have been written and organized in any number of ways to achieve the same functions. Specifically, Oracle argues that the district court erred when it: (1) concluded that each line of declaring code is uncopyrightable because the idea and expression have merged; (2) found the declaring code uncopyrightable because it employs short phrases; (3) found all aspects of the SSO devoid of protection as a "method of operation" under 17 U.S.C. § 102(b); and (4) invoked Google's "interoperability" concerns in the copyrightability analysis. For the reasons explained below, we agree with Oracle on each point.

1. Declaring Source Code

First, Oracle argues that the district court erred in concluding that each line of declaring source code is completely unprotected under the merger and short phrases doctrines. Google responds that Oracle waived its right to assert copyrightability based on the 7,000 lines of declaring code by failing "to object to instructions and a verdict form that effectively eliminated that theory from the case." Appellee Br. 67. Even if not waived, moreover, Google argues that, because there is only one way to write the names and declarations, the merger doctrine bars copyright protection.

[17] We find that Oracle did not waive arguments based on Google's literal copying of the declaring code. Prior to trial, both parties informed the court that Oracle's copyright infringement claims included the declarations of the API elements in the Android class library source code. *See* Oracle's Statement of Issues Regarding Copyright, *Oracle Am., Inc. v. Google Inc.*, No. 3:10–cv–3561 (N.D.Cal. Apr. 12, 2012), ECF No. 899–1, at 3 (Oracle accuses the "declarations of the API elements in the Android class library source code and object code that implements the 37 API packages" of copyright infringement.); *see also* Google's Proposed Statement of Issues Regarding Copyright, *Oracle Am., Inc. v. Google Inc.*, No. 3:10–cv–3561 (N.D.Cal. Apr. 12, 2012), ECF No. 901, at 2 (Oracle accuses the "declarations of the API elements in Android class library source code and object code that implements the 37 API packages.").

While Google is correct that the jury instructions and verdict form focused on the structure and organization of the packages, we agree with Oracle that there was no need for the jury to address copying of the declaring code because Google conceded that it copied it verbatim. Indeed, the district court specifically instructed the jury that "Google agrees that it uses the same names and declarations" in Android. *Final Charge to the Jury* at 10.

That the district court addressed the declaring code in its post-jury verdict copyrightability decision further confirms that the verbatim copying of declaring code remained in the case. The court explained that the "identical lines" that Google copied into Android "are those lines that specify the names, parameters and functionality of the methods and classes, lines called 'declarations' or 'headers.' " *Copyrightability Decision*, 872 F.Supp.2d at 979. The court specifically found that the declaring code was not entitled to copyright protection under the merger and short phrases doctrines. We address each in turn.

a. Merger

[18] The merger doctrine functions as an exception to the idea/expression dichotomy. It provides that, when there are a limited number of ways to express an idea, the idea is said to "merge" with its expression, and the expression becomes unprotected. *Altai*, 982 F.2d at 707–08. As noted, the Ninth Circuit treats this concept as an affirmative defense to infringement.

Ets–Hokin, 225 F.3d at 1082. Accordingly, it appears that the district court's merger analysis is irrelevant to the question of whether Oracle's API packages are copyrightable in the first instance. Regardless of when the analysis occurs, we conclude that merger does not apply on the record before us.

[19, 20] Under the merger doctrine, a court will not protect a copyrighted work from infringement if the idea contained therein can be expressed in only one way. *Satava v. Lowry*, 323 F.3d 805, 812 n. 5 (9th Cir.2003). For computer programs, "this means that when specific [parts of the code], even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to infringement." *Altai*, 982 F.2d at 708 (citation omitted). We have recognized, however, 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." *Atari*, 975 F.2d at 840.

In *Atari*, for example, Nintendo designed a program—the 10NES—to prevent its video game system from accepting unauthorized game cartridges. 975 F.2d at 836. Nintendo "chose arbitrary programming instructions and arranged them in a unique sequence to create a purely arbitrary data stream" which "serves as the key to unlock the NES." *Id.* at 840. 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.*

Here, the district court found that, "no matter how creative or imaginative a Java method specification may be, the entire world is entitled to use the same method specification (inputs, outputs, parameters) so long as the line-by-line implementations are different." *Copyrightability Decision*, 872 F.Supp.2d at 998. In its analysis, the court identified the method declaration as the idea and found that the implementation is the expression. *Id.* ("The method specification is the *idea*. The method implementation is the *expression*. No one may monopolize the *idea*.") (emphases in original). The court explained that, under the rules of Java, a programmer must use the identical "declaration or method header lines" to "declare a method specifying the *same* functionality." *Id.* at 976. Because the district court found that there was only one way to write the declaring code for each of the Java packages, it concluded that "the merger doctrine bars anyone from claiming exclusive copyright ownership" of it. *Id.* at 998. Accordingly, the court held there could be "no copyright violation in using the identical declarations." *Id.*

Google agrees with the district court that the implementing code is the expression entitled to protection—not the declaring code. Indeed, at oral argument, counsel for Google explained that, "it is not our position that none of Java is copyrightable. Obviously, Google spent two and a half years . . . to write from scratch all of the implementing code." Oral Argument at 33:16.[5] Because it is undisputed that Google wrote its own implementing code, the

5. It is undisputed that Microsoft and Apple developed mobile operating systems from scratch, using their own array of software packages. When asked whether Google could also copy all of Microsoft or Apple's declaring code—codes that obviously differ from those at issue here—counsel for Google responded: "Yes, but only the structure, sequence, and organization. Only the command structure—what you need to access the functions. You'd have to rewrite all the millions of lines of code in Apple or in Microsoft which is what Google did in Android." Oral Argument at 36:00.

copyrightability of the precise language of that code is not at issue on appeal. Instead, our focus is on the declaring code and structure of the API packages.

[21] On appeal, Oracle argues that the district court: (1) misapplied the merger doctrine; and (2) failed to focus its analysis on the options available to the original author. We agree with Oracle on both points. First, we agree that merger cannot bar copyright protection for any lines of declaring source code unless Sun/Oracle had only one way, or a limited number of ways, to write them. *See Satava*, 323 F.3d at 812 n. 5 ("Under the merger doctrine, courts will not protect a copyrighted work from infringement if the idea underlying the copyrighted work can be expressed in only one way, lest there be a monopoly on the underlying idea."). The evidence showed that Oracle had "unlimited options as to the selection and arrangement of the 7000 lines Google copied." Appellant Br. 50. Using the district court's "java.lang.Math.max" example, Oracle explains that the developers could have called it any number of things, including "Math.maximum" or "Arith.larger." This was not a situation where Oracle was selecting among preordained names and phrases to create its packages.[6] As the district court recognized, moreover, "the Android method and class names could have been different from the names of their counterparts in Java and still have worked." *Copyrightability Decision*, 872 F.Supp.2d at 976. Because "alternative expressions [we]re available," there is no merger. *See Atari*, 975 F.2d at 840.

[22] We further find that the district court erred in focusing its merger analysis on the options available to Google at the time of copying. It is well-established that copyrightability and the scope of protectable activity are to be evaluated at the time of creation, not at the time of infringement. *See Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521, 524 (9th Cir.1984) (quoting National Commission on New Technological Uses of Copyrighted Works, Final Report at 21 (1979) ("CONTU Report") (recognizing that the Copyright Act was designed "to protect all works of authorship from the moment of their fixation in any tangible medium of expression")). The focus is, therefore, on the options that were available to Sun/Oracle at the time it created the API packages. Of course, once Sun/Oracle created "java.lang.Math.max," programmers who want to use that particular package have to call it by that name. But, as the court acknowledged, nothing prevented Google from writing its own declaring code, along with its own implementing code, to achieve the same result. In such circumstances, the chosen expression simply does not merge with the idea being expressed.[7]

6. In their brief as amici curiae in support of reversal, Scott McNealy and Brian Sutphin—both former executives at Sun who were involved in the development of the Java platform—provide a detailed example of the creative choices involved in designing a Java package. Looking at the "java.text" package, they explain that it "contains 25 classes, 2 interfaces, and hundreds of methods to handle text, dates, numbers, and messages in a manner independent of natural human languages...." Br. of McNealy and Sutphin 14–15. Java's creators had to determine whether to include a java.text package in the first place, how long the package would be, what elements to include, how to organize that package, and how it would relate to other packages. *Id.* at 16. This description of Sun's creative process is consistent with the evidence presented at trial. *See* Appellant Br. 12–13 (citing testimony that it took years to write some of the Java packages and that Sun/Oracle developers had to "wrestle with what functions to include in the package, which to put in other packages, and which to omit entirely").

7. The district court did not find merger with respect to the structure, sequence, and organization of Oracle's Java API packages. Nor could it, given the court's recognition that there were myriad ways in which the API

It seems possible that the merger doctrine, when properly analyzed, would exclude the three packages identified by the district court as core packages from the scope of actionable infringing conduct. This would be so if the Java authors, at the time these packages were created, had only a limited number of ways to express the methods and classes therein if they wanted to write in the Java language. In that instance, the idea may well be merged with the expression in these three packages.[8] Google did not present its merger argument in this way below and does not do so here, however. Indeed, Google does not try to differentiate among the packages for purposes of its copyrightability analysis and does not appeal the infringement verdict as to the packages. For these reasons, we reject the trial court's merger analysis.

b. Short Phrases

[23] The district court also found that Oracle's declaring code consists of uncopyrightable short phrases. Specifically, the court concluded that, "while the Android method and class names could have been different from the names of their counterparts in Java and still have worked, copyright protection never extends to names or short phrases as a matter of law." *Copyrightability Decision*, 872 F.Supp.2d at 976.

The district court is correct that "[w]ords and short phrases such as names, titles, and slogans" are not subject to copyright protection. 37 C.F.R. § 202.1(a). The court failed to recognize, however, that the relevant question for copyrightability purposes is not whether the work at issue contains short phrases—as literary works often do—but, rather, whether those phrases are creative. *See Soc'y of Holy Transfiguration Monastery, Inc. v. Gregory*, 689 F.3d 29, 52 (1st Cir.2012) (noting that "not all short phrases will automatically be deemed uncopyrightable"); see also 1 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 2.01[B] (2013) ("[E]ven a short phrase may command copyright protection if it exhibits sufficient creativity."). And, by dissecting the individual lines of declaring code at issue into short phrases, the district court further failed to recognize that an original combination of elements can be copyrightable. *See Softel, Inc. v. Dragon Med. & Scientific Commc'ns*, 118 F.3d 955, 964 (2d Cir.1997) (noting that, in Feist, "the Court made quite clear that a compilation of nonprotectible elements can enjoy copyright protection even though its constituent elements do not").

packages could have been organized. Indeed, the court found that the SSO is original and that ''nothing in the rules of the Java language ... required that Google replicate the same groupings.'' *Copyrightability Decision*, 872 F.Supp.2d at 999. As discussed below, however, the court nonetheless found that the SSO is an uncopyrightable ''method of operation.''

8. At oral argument, counsel for Oracle was asked whether we should view the three core packages ''differently vis-à-vis the concept of a method of operation than the other packages.'' *See* Oral Argument at 7:43. He responded: ''I think not your Honor. I would view them differently with respect to fair use.... It's not that they are more basic. It's that there are just several methods, that is, routines, within just those three packages that are necessary to 'speak the Java language.' Nothing in the other thirty-four packages is necessary in order to speak in Java, so to speak.'' *Id.* Counsel conceded, however, that this issue ''might go to merger. It might go to the question whether someone—since we conceded that it's okay to use the language—if it's alright to use the language that there are certain things that the original developers had to say in order to use that language, arguably, although I still think it's really a fair use analysis.'' *Id.*

By analogy, 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. The question is not whether a short phrase or series of short phrases can be extracted from the work, but whether the manner in which they are used or strung together exhibits creativity.

Although the district court apparently focused on individual lines of code, Oracle is not seeking copyright protection for a specific short phrase or word. Instead, the portion of declaring code at issue is 7,000 lines, and Google's own "Java guru" conceded that there can be "creativity and artistry even in a single method declaration." Joint Appendix ("J.A.") 20,970. Because Oracle "exercised creativity in the selection and arrangement" of the method declarations when it created the API packages and wrote the relevant declaring code, they contain protectable expression that is entitled to copyright protection. *See Atari*, 975 F.2d at 840; *see also* 17 U.S.C. §§ 101, 103 (recognizing copyright protection for "compilations" which are defined as work that is "selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship"). Accordingly, we conclude that the district court erred in applying the short phrases doctrine to find the declaring code not copyrightable.

c. Scenes a Faire

[24–26] The scenes a faire doctrine, which is related to the merger doctrine, operates to bar certain otherwise creative expression from copyright protection. *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1444 (9th Cir.1994). It provides that "expressive elements of a work of authorship are not entitled to protection against infringement if they are standard, stock, or common to a topic, or if they necessarily follow from a common theme or setting." *Mitel*, 124 F.3d at 1374. Under this doctrine, "when certain commonplace expressions are indispensable and naturally associated with the treatment of a given idea, those expressions are treated like ideas and therefore [are] not protected by copyright." *Swirsky v. Carey*, 376 F.3d 841, 850 (9th Cir.2004). In the computer context, "the scene a faire doctrine denies protection to program elements that are dictated by external factors such as 'the mechanical specifications of the computer on which a particular program is intended to run' or 'widely accepted programming practices within the computer industry.' " *Softel*, 118 F.3d at 963 (citation omitted).

The trial court rejected Google's reliance on the scenes a faire doctrine. It did so in a footnote, finding that Google had failed to present evidence to support the claim that either the grouping of methods within the classes or the code chosen for them "would be so expected and customary as to be permissible under the scenes a faire doctrine." *Copyrightability Decision*, 872 F.Supp.2d at 999 n. 9. Specifically, the trial court found that "it is impossible to say on this record that *all* of the classes and their contents are typical of such classes and, on this record, this order rejects Google's global argument based on *scenes a faire*." *Id.*

On appeal, Google refers to scenes a faire concepts briefly, as do some amici, apparently contending that, because programmers have become accustomed to and comfortable using the groupings in the Java API packages, those groupings are so commonplace as to be indispensable to the expression of an acceptable programming platform. As such, the argument goes, they are so associated with the "idea" of what the packages are accomplishing that they should be treated as ideas rather than

expression. *See* Br. of Amici Curiae Rackspace US, Inc., et al. at 19–22.

[27] Google cannot rely on the scenes a faire doctrine as an alternative ground upon which we might affirm the copyrightability judgment of the district court. This is so for several reasons. First, as noted, like merger, in the Ninth Circuit, the scenes a faire doctrine is a component of the infringement analysis. "[S]imilarity of expression, whether literal or non-literal, which necessarily results from the fact that the common idea is only capable of expression in more or less stereotyped form, will preclude a finding of actionable similarity." 4 Nimmer on Copyright § 13.03[B][3]. Thus, the expression is not excluded from copyright protection; it is just that certain copying is forgiven as a necessary incident of *any* expression of the underlying idea. *See Satava*, 323 F.3d at 810 n. 3 ("The Ninth Circuit treats scenes a faire as a defense to infringement rather than as a barrier to copyrightability.").

Second, Google has not objected to the trial court's conclusion that Google failed to make a sufficient factual record to support its contention that the groupings and code chosen for the 37 Java API packages were driven by external factors or premised on features that were either commonplace or essential to the idea being expressed. Google provides no record citations indicating that such a showing was made and does not contend that the trial court erred when it expressly found it was not. Indeed, Google does not even make this argument with respect to the core packages.

[28] Finally, Google's reliance on the doctrine below and the amici reference to it here are premised on a fundamental misunderstanding of the doctrine. Like merger, the focus of the scenes a faire doctrine is on the circumstances presented to the creator, not the copier. *See Mitel*, 124 F.3d at 1375 (finding error to the extent the trial court discussed "whether external factors such as market forces and efficiency considerations justified Iqtel's copying of the command codes"). The court's analytical focus must be upon the external factors that dictated Sun's selection of classes, methods, and code—not upon what Google encountered at the time it chose to copy those groupings and that code. *See id.* "[T]he scenes a faire doctrine identifies and excludes from protection against infringement expression whose creation 'flowed naturally from considerations external to the author's creativity.' " *Id.* (quoting Nimmer § 13.03[F][3], at 13–131 (1997)). It is this showing the trial court found Google failed to make, and Google cites to nothing in the record which indicates otherwise.

For these reasons, the trial court was correct to conclude that the scenes a faire doctrine does not affect the copyrightability of either the declaring code in, or the SSO of, the Java API packages at issue.

2. The Structure, Sequence, and Organization of the API Packages

The district court found that the SSO of the Java API packages is creative and original, but nevertheless held that it is a "system or method of operation ... and, therefore, cannot be copyrighted" under 17 U.S.C. § 102(b). *Copyrightability Decision*, 872 F.Supp.2d at 976–77. In reaching this conclusion, the district court seems to have relied upon language contained in a First Circuit decision: *Lotus Development Corp. v. Borland International, Inc.*, 49 F.3d 807 (1st Cir.1995), *aff'd without opinion by equally divided court*, 516 U.S. 233, 116 S.Ct. 804, 133 L.Ed.2d 610 (1996).[9]

9. The Supreme Court granted certiorari in *Lotus*, but, shortly after oral argument, the Court announced that it was equally divided and that Justice Stevens took no part in the

In *Lotus*, it was undisputed that the defendant copied the menu command hierarchy and interface from Lotus 1–2–3, a computer spreadsheet program "that enables users to perform accounting functions electronically on a computer." 49 F.3d at 809. The menu command hierarchy referred to a series of commands—such as "Copy," "Print," and "Quit"—which were arranged into more than 50 menus and submenus. *Id.* Although the defendant did not copy any Lotus source code, it copied the menu command hierarchy into its rival program. The question before the court was "whether a computer menu command hierarchy is copyrightable subject matter." *Id.*

Although it accepted the district court's finding that Lotus developers made some expressive choices in selecting and arranging the command terms, the First Circuit found that the command hierarchy was not copyrightable because, among other things, it was a "method of operation" under Section 102(b). In reaching this conclusion, the court defined a "method of operation" as "the means by which a person operates something, whether it be a car, a food processor, or a computer." *Id.* at 815.[10] Because the Lotus menu command hierarchy provided "the means by which users control and operate Lotus 1–2–3," it was deemed unprotectable. *Id.* For example, if users wanted to copy material, they would use the "Copy" command and the command terms would tell the computer what to do. According to the *Lotus* court, the "fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the question of whether it is a 'method of operation.' " *Id.* at 816. (noting that "our initial inquiry is not whether the Lotus menu command hierarchy incorporates any expression"). The court further indicated that, "[i]f specific words are essential to operating something, then they are part of a 'method of operation' and, as such, are unprotectable." *Id.*

On appeal, Oracle argues that the district court's reliance on *Lotus* is misplaced because it is distinguishable on its facts and is inconsistent with Ninth Circuit law. We agree. First, while the defendant in *Lotus* did not copy any of the underlying code, Google concedes that it copied portions of Oracle's declaring source code verbatim. Second, the *Lotus* court found that the commands at issue there (copy, print, etc.) were not creative, but it is undisputed here that the declaring code and the structure and organization of the API packages are both creative and original. Finally, while the court in *Lotus* found the commands at issue were "essential to operating" the system, it is undisputed that—other than perhaps as to the three core packages—Google did not need to copy the structure, sequence, and organization of the Java API packages to write programs in the Java language.

More importantly, however, the Ninth Circuit has not adopted the court's "method of operation" reasoning in *Lotus*, and we conclude that it is inconsistent with binding precedent.[11] Specifically, we find

consideration or decision of the case. The Court therefore left the First Circuit's decision undisturbed. *See Lotus*, 516 U.S. at 233–34, 116 S.Ct. 804.

10. The *Lotus* majority cited no authority for this definition of "method of operation."

11. As Oracle points out, the Ninth Circuit has cited *Lotus* only one time, on a procedural issue. *See Danjaq LLC v. Sony Corp.*, 263 F.3d 942, 954 (9th Cir.2001) (citing *Lotus* for the proposition that delay "has been held permissible, among other reasons, when it is necessitated by the exhaustion of remedies through the administrative process . . . when it is used to evaluate and prepare a complicated claim").

that *Lotus* is inconsistent with Ninth Circuit case law recognizing that the structure, sequence, and organization of a computer program is eligible for copyright protection where it qualifies as an expression of an idea, rather than the idea itself. *See Johnson Controls,* 886 F.2d at 1175–76. And, while the court in *Lotus* held "that expression that is part of a 'method of operation' cannot be copyrighted," 49 F.3d at 818, this court—applying Ninth Circuit law—reached the exact opposite conclusion, finding that copyright protects "the expression of [a] process or method," *Atari,* 975 F.2d at 839.

We find, moreover, that the hard and fast rule set down in *Lotus* and employed by the district court here—i.e., that elements which perform a function can never be copyrightable—is at odds with the Ninth Circuit's endorsement of the abstraction-filtration-comparison analysis discussed earlier. As the Tenth Circuit concluded in expressly rejecting the *Lotus* "method of operation" analysis, in favor of the Second Circuit's abstraction-filtration-comparison test, "although an element of a work may be characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection." *Mitel,* 124 F.3d at 1372. Specifically, the court found that Section 102(b) "does not extinguish the protection accorded a particular expression of an idea merely because that expression is embodied in a method of operation at a higher level of abstraction." *Id.*

Other courts agree that components of a program that can be characterized as a "method of operation" may nevertheless be copyrightable. For example, the Third Circuit rejected a defendant's argument that operating system programs are "per se" uncopyrightable because an operating system is a "method of operation" for a computer. *Apple Computer, Inc. v. Franklin Computer Corp.,* 714 F.2d 1240, 1250–52 (3d Cir.1983). The court distinguished between the "method which instructs the computer to perform its operating functions" and "the instructions themselves," and found that the instructions were copyrightable. *Id.* at 1250–51. In its analysis, the court noted: "[t]hat the words of a program are used ultimately in the implementation of a process should in no way affect their copyrightability." *Id.* at 1252 (quoting CONTU Report at 21). The court 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." *Id.* at 1253. Notably, no other circuit has adopted the First Circuit's "method of operation" analysis.

Courts have likewise found that classifying a work as a "system" does not preclude copyright for the particular expression of that system. *See Toro Co. v. R & R Prods. Co.,* 787 F.2d 1208, 1212 (8th Cir. 1986) (rejecting the district court's decision that "appellant's parts numbering system is not copyrightable because it is a 'system' " and indicating that Section 102(b) does not preclude protection for the "particular expression" of that system); *see also Am. Dental Ass'n v. Delta Dental Plans Ass'n,* 126 F.3d 977, 980 (7th Cir. 1997) ("A dictionary cannot be called a 'system' just because new novels are written using words, all of which appear in the dictionary. Nor is word-processing software a 'system' just because it has a command structure for producing paragraphs.").

Here, the district court recognized that the SSO "resembles a taxonomy," but found that "it is nevertheless a command structure, a system or method of operation—a long hierarchy of over six thousand

commands to carry out pre-assigned functions." *Copyrightability Decision*, 872 F.Supp.2d at 999–1000.[12] In other words, the court concluded that, although the SSO is expressive, it is not copyrightable because it is also functional. The problem with the district court's approach is that computer programs are by definition functional—they are all designed to accomplish some task. Indeed, the statutory definition of "computer program" acknowledges that they function "to bring about a certain result." *See* 17 U.S.C. § 101 (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"). If we were to accept the district court's suggestion that a computer program is uncopyrightable simply because it "carr[ies] out preassigned functions," no computer program is protectable. That result contradicts Congress's express intent to provide copyright protection to computer programs, as well as binding Ninth Circuit case law finding computer programs copyrightable, despite their utilitarian or functional purpose. Though the trial court did add the caveat that it "does not hold that the structure, sequence and organization of all computer programs may be stolen," *Copyrightability Decision*, 872 F.Supp.2d at 1002, it is hard to see how its method of operation analysis could lead to any other conclusion.

[29–31] While it does not appear that the Ninth Circuit has addressed the precise issue, we conclude that a set of commands to instruct a computer to carry out desired operations may contain expression that is eligible for copyright protection. *See Mitel*, 124 F.3d at 1372. 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. Section 102(b) does not, as Google seems to suggest, automatically deny copyright protection to elements of a computer program that are functional. Instead, as noted, Section 102(b) codifies the idea/expression dichotomy and the legislative history confirms that, among other things, Section 102(b) was "intended to make clear that the expression adopted by the programmer is the copyrightable element in a computer program." H.R.Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5670. Therefore, even if an element directs a computer to perform operations, the court must nevertheless determine whether it contains any separable expression entitled to protection.

On appeal, Oracle does not—and concedes that it cannot—claim copyright in the idea of organizing functions of a computer program or in the "package-class-method" organizational structure in the abstract. Instead, Oracle claims copyright protection only in its *particular* way of naming and organizing each of the 37 Java API packages.[13] Oracle recognizes, for example, 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

12. This analogy by the district court is meaningful because taxonomies, in varying forms, have generally been deemed copyrightable. *See, e.g., Practice Mgmt. Info. Corp. v. Am. Med. Ass'n,* 121 F.3d 516, 517–20 (9th Cir. 1997); *Am. Dental,* 126 F.3d at 978–81.

13. At oral argument, counsel for Oracle explained that it "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." Oral Argument at 16:44.

achieve the same function." Appellant Reply Br. 13–14 (citation omitted). Thus, Oracle concedes that Google and others could employ the Java language—much like anyone could employ the English language to write a paragraph without violating the copyrights of other English language writers. And, that Google may employ the "package-class-method" structure much like authors can employ the same rules of grammar chosen by other authors without fear of infringement. What Oracle contends is that, beyond that point, Google, like any author, is not permitted to employ the precise phrasing or precise structure chosen by Oracle to flesh out the substance of its packages—the details and arrangement of the prose.

As the district court acknowledged, Google could have structured Android differently and could have chosen different ways to express and implement the functionality that it copied.[14] Specifically, the court found that "the very same functionality could have been offered in Android without duplicating the exact command structure used in Java." *Copyrightability Decision*, 872 F.Supp.2d at 976. The court further explained that Google could have offered the same functions in Android by "rearranging the various methods under different groupings among the various classes and packages." *Id.* The evidence showed, moreover, that Google designed many of its own API packages from scratch, and, thus, could have designed its own corresponding 37 API packages if it wanted to do so.

[32] Given the 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.

3. Google's Interoperability Arguments are Irrelevant to Copyrightability

Oracle also argues that the district court erred in invoking interoperability in its copyrightability analysis. Specifically, Oracle argues that Google's interoperability arguments are only relevant, if at all, to fair use—not to the question of whether the API packages are copyrightable. We agree.

In characterizing the SSO of the Java API packages as a "method of operation," the district court explained that "[d]uplication of the command structure is necessary for interoperability." *Copyrightability Decision*, 872 F.Supp.2d at 977. The court found that, "[i]n order for at least some of [the pre-Android Java] code to run on Android, Google was required to provide the same java.package.Class.method( ) command system using the same names with the same 'taxonomy' and with the same functional specifications." *Id.* at

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." Br. of McNealy and Sutphin 17. For example, in Java, a developer setting the time zone would call the "setTimeZone" method within the "DateFormat" class of the java.text package. *Id.* Apple's iOS platform, on the other hand, "devotes an entire class to set the time zone in an application—the 'NSTimeZone' class" which is in the "Foundation framework." *Id.* at 17–18 (noting that a "framework is Apple's terminology for a structure conceptually similar to Java's 'package' "). Microsoft provides similar functionality with "an entirely different structure, naming scheme, and selection." *Id.* at 18 ("In its Windows Phone development platform, Microsoft stores its time zone programs in the 'TimeZoneInfo' class in its 'Systems' namespace (Microsoft's version of a 'package' or 'framework')."). Again, this is consistent with the evidence presented at trial.

1000 (emphasis omitted). And, the court concluded that "Google replicated what was necessary to achieve a degree of interoperability—but no more, taking care, as said before, to provide its own implementations." *Id.* In reaching this conclusion, the court relied primarily on two Ninth Circuit decisions: *Sega Enterprises v. Accolade, Inc.,* 977 F.2d 1510 (9th Cir.1992), and *Sony Computer Entertainment, Inc. v. Connectix, Corp.,* 203 F.3d 596 (9th Cir. 2000).

Both *Sega* and *Sony* are fair use cases in which copyrightability was addressed only tangentially. In *Sega,* for example, Sega manufactured a video game console and game cartridges that contained hidden functional program elements necessary to achieve compatibility with the console. Defendant Accolade: (1) reverse-engineered Sega's video game programs to discover the requirements for compatibility; and (2) created its own games for the Sega console. *Sega,* 977 F.2d at 1514–15. As part of the reverse-engineering process, Accolade made intermediate copies of object code from Sega's console. *Id.* Although the court recognized that the intermediate copying of computer code may infringe Sega's copyright, it concluded that "disassembly of copyrighted object code is, as a matter of law, a fair use of the copyrighted work if such disassembly provides the only means of access to those elements of the code that are not protected by copyright and the copier has a legitimate reason for seeking such access." *Id.* at 1518. The court agreed with Accolade that its copying was necessary to examine the unprotected functional aspects of the program. *Id.* at 1520. And, because Accolade had a legitimate interest in making its cartridges compatible with Sega's console, the court found that Accolade's intermediate copying was fair use.

Likewise, in *Sony,* the Ninth Circuit found that the defendant's reverse engineering and intermediate copying of Sony's copyrighted software program "was a fair use for the purpose of gaining access to the unprotected elements of Sony's software." *Sony,* 203 F.3d at 602. The court explained that Sony's software program contained unprotected functional elements and that the defendant could only access those elements through reverse engineering. *Id.* at 603. The defendant used that information to create a software program that let consumers play games designed for Sony's PlayStation console on their computers. Notably, the defendant's software program did not contain any of Sony's copyrighted material. *Id.* at 598.

The district court characterized *Sony* and *Sega* as "close analogies" to this case. *Copyrightability Decision,* 872 F.Supp.2d at 1000. According to the court, both decisions "held that interface procedures that were necessary to duplicate in order to achieve interoperability were functional aspects not copyrightable under Section 102(b)." *Id.* The district court's reliance on *Sega* and *Sony* in the copyrightability context is misplaced, however.

As noted, both cases were focused on fair use, not copyrightability. In *Sega,* for example, the only question was whether Accolade's intermediate copying was fair use. The court never addressed the question of whether Sega's software code, which had functional elements, also contained separable creative expression entitled to protection. Likewise, although the court in *Sony* determined that Sony's computer program had functional elements, it never addressed whether it also had expressive elements. *Sega* and *Sony* are also factually distinguishable because the defendants in those cases made intermediate copies to understand the functional aspects of the copyrighted works and then created new products. *See Sony,* 203 F.3d at 606–07; *Sega,* 977 F.2d at 1522–23. This is not

a case where Google reverse-engineered Oracle's Java packages to gain access to unprotected functional elements contained therein. As the former Register of Copyrights of the United States pointed out in his brief amicus curiae, "[h]ad Google reverse engineered the programming packages to figure out the ideas and functionality of the original, and then created its own structure and its own literal code, Oracle would have no remedy under copyright whatsoever." Br. for Amicus Curiae Ralph Oman 29. Instead, Google chose to copy both the declaring code and the overall SSO of the 37 Java API packages at issue.

We disagree with Google's suggestion that *Sony* and *Sega* created an "interoperability exception" to copyrightability. *See* Appellee Br. 39 (citing *Sony* and *Sega* for the proposition that "compatibility elements are not copyrightable under section 102(b)" (emphasis omitted)). Although both cases recognized that the software programs at issue there contained unprotected functional elements, a determination that some elements are unprotected is not the same as saying that the entire work loses copyright protection. To accept Google's reading would contradict Ninth Circuit case law recognizing that both the literal and non-literal components of a software program are eligible for copyright protection. *See Johnson Controls*, 886 F.2d at 1175. And it would ignore the fact that the Ninth Circuit endorsed the abstraction-filtration-comparison inquiry in *Sega* itself.

As previously discussed, a court must examine the software program to determine whether it contains creative expression that can be separated from the underlying function. *See Sega*, 977 F.2d at 1524–25. In doing so, the court filters out the elements of the program that are "ideas" as well as elements that are "dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself." *Altai*, 982 F.2d at 707.

[33] To determine "whether certain aspects of an allegedly infringed software are not protected by copyright law, the focus is on external factors that influenced the choice of the creator of the infringed product." *Dun & Bradstreet Software Servs., Inc. v. Grace Consulting, Inc.*, 307 F.3d 197, 215 (3d Cir.2002) (citing *Altai*, 982 F.2d at 714; *Mitel*, 124 F.3d at 1375). The Second Circuit, for example, has noted that programmers are often constrained in their design choices by "extrinsic considerations" including "the mechanical specifications of the computer on which a particular program is intended to run" and "compatibility requirements of other programs with which a program is designed to operate in conjunction." *Altai*, 982 F.2d at 709–10 (citing 3 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 13.01 at 13–66–71 (1991)). The Ninth Circuit has likewise recognized that: (1) computer programs "contain many logical, structural, and visual display elements that are dictated by ... external factors such as compatibility requirements and industry demands"; and (2) "[i]n some circumstances, even the exact set of commands used by the programmer is deemed functional rather than creative for purposes of copyright." *Sega*, 977 F.2d at 1524 (internal citation omitted).

[34] Because copyrightability is focused on the choices available to the plaintiff at the time the computer program was created, the relevant compatibility inquiry asks whether the plaintiff's choices were dictated by a need to ensure that its program worked with existing third-party programs. *Dun & Bradstreet*, 307 F.3d at 215; *see also Atari*, 975 F.2d at 840 ("External factors did not dictate the design of the 10NES program."). Whether a defen-

dant later seeks to make its program interoperable with the plaintiff's program has no bearing on whether the software the plaintiff created had any design limitations dictated by external factors. *See Dun & Bradstreet*, 307 F.3d at 215 (finding an expert's testimony on interoperability "wholly misplaced" because he "looked at externalities from the eyes of the plagiarist, not the eyes of the program's creator"). Stated differently, the focus is on the compatibility needs and programming choices of the party claiming copyright protection—not the choices the defendant made to achieve compatibility with the plaintiff's program. Consistent with this approach, courts have recognized that, once the plaintiff creates a copyrightable work, 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." *Apple Computer*, 714 F.2d at 1253.

[35] Given this precedent, we conclude that the district court erred in focusing its interoperability analysis on Google's desires for its Android software. *See Copyrightability Decision*, 872 F.Supp.2d at 1000 ("Google replicated what was necessary to achieve a degree of interoperability" with Java.). Whether Google's software is "interoperable" in some sense with any aspect of the Java platform (although as Google concedes, certainly not with the JVM) has no bearing on the threshold question of whether Oracle's software is copyrightable. It is the interoperability and other needs of Oracle—not those of Google—that apply in the copyrightability context, and there is no evidence that when Oracle created the Java API packages at issue it did so to meet compatibility requirements of other pre-existing programs.

Google maintains on appeal that its use of the "Java class and method names and declarations was 'the only and essential means' of achieving a degree of interoperability with existing programs written in the [Java language]." Appellee Br. 49. Indeed, given the record evidence that Google designed Android so that it would *not* be compatible with the Java platform, or the JVM specifically, we find Google's interoperability argument confusing. While Google repeatedly cites to the district court's finding that Google had to copy the packages so that an app written in Java could run on Android, it cites to no evidence in the record that any such app exists and points to no Java apps that either pre-dated or post-dated Android that could run on the Android platform.[15] The compatibility Google sought to foster was not with Oracle's Java platform or with the JVM central to that platform. Instead, Google wanted to capitalize on the fact that software developers were already trained and experienced in using the Java API packages at issue. The district court agreed, finding that, as to the 37 Java API packages, "Google believed Java application programmers would want to find the same 37 sets of functionalities in the new Android system callable by the same names as used in Java." *Copyrightability Decision*, 872 F.Supp.2d at 978. Google's interest was in accelerating its development process by "leverag[ing] Java for its existing base of developers." J.A.2033,

15. During oral argument, Google's counsel stated that "a program written in the Java language can run on Android if it's only using packages within the 37. So if I'm a developer and I have written a program, I've written it in Java, I can stick an Android header on it and it will run in Android because it is using the identical names of the classes, methods, and packages." Oral Argument at 31:31. Counsel did not identify any programs that use only the 37 API packages at issue, however, and did not attest that any such program would be useful. Nor did Google cite to any record evidence to support this claim.

2092. Although this competitive objective might be relevant to the fair use inquiry, we conclude that it is irrelevant to the copyrightability of Oracle's declaring code and organization of the API packages.

[36, 37] Finally, to the extent Google suggests that it was entitled to copy the Java API packages because they had become the effective industry standard, we are unpersuaded. Google cites no authority for its suggestion that copyrighted works lose protection when they become popular, and we have found none.[16] In fact, the Ninth Circuit has rejected the argument that a work that later becomes the industry standard is uncopyrightable. *See Practice Mgmt. Info. Corp. v. Am. Med. Ass'n,* 121 F.3d 516, 520 n. 8 (9th Cir.1997) (noting that the district court found plaintiff's medical coding system entitled to copyright protection, and that, although the system had become the industry standard, plaintiff's copyright did not prevent competitors "from developing comparative or better coding systems and lobbying the federal government and private actors to adopt them. It simply prevents wholesale copying of an existing system."). Google was free to develop its own API packages and to "lobby" programmers to adopt them. Instead, it chose to copy Oracle's declaring code and the SSO to capitalize on the preexisting community of programmers who were accustomed to using the Java API packages. That desire has nothing to do with copyrightability. For these reasons, we find that Google's industry standard argument has no bearing on the copyrightability of Oracle's work.

B. Fair Use

As noted, the jury hung on Google's fair use defense, and the district court declined to order a new trial given its conclusion that the code and structure Google copied were not entitled to copyright protection. On appeal, Oracle argues that: (1) a remand to decide fair use "is pointless"; and (2) this court should find, as a matter of law, that "Google's commercial use of Oracle's work in a market where Oracle already competed was not fair use." Appellant Br. 68.

[38] Fair use is an affirmative defense to copyright infringement and is codified in Section 107 of the Copyright Act. *Golan,* 132 S.Ct. at 890 ("[T]he fair use defense, is codified at 17 U.S.C. § 107."). Section 107 permits use of copyrighted work if it is "for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research." 17 U.S.C. § 107. The fair use doctrine has been referred to as " 'the most troublesome in the whole law of copyright.' " *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)). It both permits and requires "courts to avoid rigid application of the copyright statute

16. Google argues that, in the same way a formerly distinctive trademark can become generic over time, a program element can lose copyright protection when it becomes an industry standard. But "it is to be expected that phrases and other fragments of expression in a highly successful copyrighted work will become part of the language. That does not mean they lose all protection in the manner of a trade name that has become generic." *Warner Bros., Inc. v. Am. Broadcasting Cos.,* 720 F.2d 231, 242 (2d Cir.1983) ("No matter how well known a copyrighted phrase becomes, its author is entitled to guard against its appropriation to promote the sale of commercial products."). Notably, even when a patented method or system becomes an acknowledged industry standard with acquiescence of the patent owner, any permissible use generally requires payment of a reasonable royalty, which Google refused to do here. *See generally In re Innovatio IP Ventures, LLC,* No. 11–C–9308, 2013 U.S. Dist. LEXIS 144061 (N.D.Ill. Sept. 27, 2013).

when, on occasion, it would stifle the very creativity which that law is designed to foster." *Campbell v. Acuff–Rose Music, Inc.*, 510 U.S. 569, 577, 114 S.Ct. 1164, 127 L.Ed.2d 500 (1994) (quoting *Stewart v. Abend*, 495 U.S. 207, 236, 110 S.Ct. 1750, 109 L.Ed.2d 184 (1990)).

[39] "Section 107 requires a case-by-case determination whether a particular use is fair, and the statute notes four nonexclusive factors to be considered." *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 549, 105 S.Ct. 2218, 85 L.Ed.2d 588 (1985). Those factors are: (1) "the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;" (2) "the nature of the copyrighted work;" (3) "the amount and substantiality of the portion used in relation to the copyrighted work as a whole;" and (4) "the effect of the use upon the potential market for or value of the copyrighted work." 17 U.S.C. § 107. The Supreme Court has explained that all of the statutory factors "are to be explored, and the results weighed together, in light of the purpose[ ] of copyright," which is "[t]o promote the Progress of Science and useful Arts." *Campbell*, 510 U.S. at 578, 575, 114 S.Ct. 1164 (internal citations omitted).

[40, 41] "Fair use is a mixed question of law and fact." *Harper & Row*, 471 U.S. at 560, 105 S.Ct. 2218. Thus, while subsidiary and controverted findings of fact must be reviewed for clear error under Rule 52 of the Federal Rules of Civil Procedure, the Ninth Circuit reviews the ultimate application of those facts de novo. *See Seltzer v. Green Day, Inc.*, 725 F.3d 1170, 1175 (9th Cir.2013) (citing *SOFA Entm't, Inc. v. Dodger Prods., Inc.*, 709 F.3d 1273, 1277 (9th Cir.2013)). Where there are no material facts at issue and "the parties dispute only the ultimate conclusions to be drawn from those facts, we may draw those conclusions without usurping the function of the jury." *Id.* (citing *Fisher v. Dees*, 794 F.2d 432, 436 (9th Cir.1986)). Indeed, the Supreme Court has specifically recognized that, "[w]here the district court has found facts sufficient to evaluate each of the statutory factors, an appellate court 'need not remand for further factfinding ... [but] may conclude as a matter of law that [the challenged use] [does] not qualify as a fair use of the copyrighted work.' " *Harper & Row*, 471 U.S. at 560, 105 S.Ct. 2218 (citation omitted).

[42] Of course, the corollary to this point is true as well—where there are material facts in dispute and those facts have not yet been resolved by the trier of fact, appellate courts may not make findings of fact in the first instance. *See Shawmut Bank, N.A. v. Kress Assocs.*, 33 F.3d 1477, 1504 (9th Cir.1994) ("[W]e must avoid finding facts in the first instance."); *see also Golden Bridge Tech., Inc. v. Nokia, Inc.*, 527 F.3d 1318, 1323 (Fed.Cir. 2008) ("Appellate courts review district court judgments; we do not find facts."). Here, it is undisputed that neither the jury nor the district court made findings of fact to which we can refer in assessing the question of whether Google's use of the API packages at issue was a "fair use" within the meaning of Section 107. Oracle urges resolution of the fair use question by arguing that the trial court should have decided the question as a matter of law based on the undisputed facts developed at trial, and that we can do so as well. Google, on the other hand, argues that many critical facts regarding fair use are in dispute. It asserts that the fact that the jury could not reach a resolution on the fair use defense indicates that at least some presumably reasonable jurors found its use to be fair. And, Google asserts that, even if it is true that the district court erred in discussing concepts of "interoperability" when considering copyrightability, those

concepts are still relevant to its fair use defense. We turn first to a more detailed examination of fair use.

[43] The first factor in the fair use inquiry involves "the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes." 17 U.S.C. § 107(1). This factor involves two subissues: (1) "whether and to what extent the new work is transformative," *Campbell*, 510 U.S. at 579, 114 S.Ct. 1164 (citation and internal quotation marks omitted); and (2) whether the use serves a commercial purpose.

[44] A use is "transformative" if it "adds something new, with a further purpose or different character, altering the first with new expression, meaning or message." *Id.* The critical question is "whether the new work merely supersede[s] the objects of the original creation ... or instead adds something new." *Id.* (citations and internal quotation marks omitted). This inquiry "may be guided by the examples given in the preamble to § 107, looking to whether the use is for criticism, or comment, or news reporting, and the like." *Id.* at 578–79, 114 S.Ct. 1164. "The Supreme Court has recognized that parodic works, like other works that comment and criticize, are by their nature often sufficiently transformative to fit clearly under the fair use exception." *Mattel, Inc. v. Walking Mountain Prods.*, 353 F.3d 792, 800 (9th Cir.2003) (citing *Campbell*, 510 U.S. at 579, 114 S.Ct. 1164).

[45] Courts have described new works as "transformative" when "the works use copy-righted material for purposes distinct from the purpose of the original material." *Elvis Presley Enters., Inc. v. Passport Video*, 349 F.3d 622, 629 (9th Cir.2003) ("Here, Passport's use of many of the television clips is transformative because they are cited as historical reference points in the life of a remarkable entertainer."), *overruled on other grounds by Flexible Lifeline Sys., Inc. v. Precision Lift, Inc.*, 654 F.3d 989, 995 (9th Cir.2011) (per curiam); *see also Bouchat v. Baltimore Ravens Ltd. P'ship*, 619 F.3d 301, 309–10 (4th Cir.2010) (quoting *A.V. ex rel. Vanderhye v. iParadigms, LLC*, 562 F.3d 630, 638 (4th Cir.2009) ("[A] transformative use is one that 'employ[s] the quoted matter in a different manner or for a different purpose from the original.' ")). "A use is considered transformative only where a defendant changes a plaintiff's copyrighted work or uses the plaintiff's copyrighted work in a different context such that the plaintiff's work is transformed into a new creation." *Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146, 1165 (9th Cir.2007) (quoting *Wall Data Inc. v. L.A. County Sheriff's Dep't*, 447 F.3d 769, 778 (9th Cir.2006), and finding that Google's use of thumbnail images in its search engine was "highly transformative").

[46, 47] A work is not transformative where the user "makes no alteration to the *expressive content or message* of the original work." *Seltzer*, 725 F.3d at 1177; *see also Wall Data*, 447 F.3d at 778 ("The Sheriff's Department created exact copies of RUMBA's software. It then put those copies to the identical purpose as the original software. Such a use cannot be considered transformative."); *Monge*, 688 F.3d at 1176 (finding that a magazine's publication of photographs of a secret celebrity wedding "sprinkled with written commentary" was "at best minimally transformative" where the magazine "did not transform the photos into a new work ... or incorporate the photos as part of a broader work"); *Elvis Presley Enters.*, 349 F.3d at 629 (finding that use of copyrighted clips of Elvis's television appearances was not transformative where "some of the clips [we]re played without much interruption, if any ... [and] instead serve[d] the

same intrinsic entertainment value that is protected by Plaintiffs' copyrights."). Where the use "is for the same intrinsic purpose as [the copyright holder's] ... such use seriously weakens a claimed fair use." *Worldwide Church of God v. Phila. Church of God, Inc.,* 227 F.3d 1110, 1117 (9th Cir.2000) (quoting *Weissmann v. Freeman,* 868 F.2d 1313, 1324 (2d Cir. 1989)).

[48, 49] Analysis of the first factor also requires inquiry into the commercial nature of the use. Use of the copyrighted work that is commercial "tends to weigh against a finding of fair use." *Harper & Row,* 471 U.S. at 562, 105 S.Ct. 2218 ("The crux of the profit/nonprofit distinction is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material without paying the customary price."). "[T]he more transformative the new work, the less will be the significance of other factors, like commercialism, that may weigh against a finding of fair use." *Campbell,* 510 U.S. at 579, 114 S.Ct. 1164.

[50–52] The second factor—the nature of the copyrighted work—"calls for recognition that some works are closer to the core of intended copyright protection than others, with the consequence that fair use is more difficult to establish when the former works are copied." *Id.* at 586, 114 S.Ct. 1164. This factor "turns on whether the work is informational or creative." *Worldwide Church of God,* 227 F.3d at 1118; *see also Harper & Row,* 471 U.S. at 563, 105 S.Ct. 2218 ("The law generally recognizes a greater need to disseminate factual works than works of fiction or fantasy."). Creative expression "falls within the core of the copyright's protective purposes." *Campbell,* 510 U.S. at 586, 114 S.Ct. 1164. Because computer programs have both functional and expressive components, however, where the functional components are themselves unprotected (because, e.g., they are dictated by considerations of efficiency or other external factors), those elements should be afforded "a lower degree of protection than more traditional literary works." *Sega,* 977 F.2d at 1526. Thus, where the nature of the work is such that purely functional elements exist in the work and it is necessary to copy the expressive elements in order to perform those functions, consideration of this second factor arguably supports a finding that the use is fair.

[53–55] The third factor asks the court to examine "the amount and substantiality of the portion used in relation to the copyrighted work as a whole." 17 U.S.C. § 107(3). Analysis of this factor is viewed in the context of the copyrighted work, not the infringing work. Indeed, the statutory language makes clear that "a taking may not be excused merely because it is insubstantial with respect to the *infringing* work." *Harper & Row,* 471 U.S. at 565, 105 S.Ct. 2218. "As Judge Learned Hand cogently remarked, 'no plagiarist can excuse the wrong by showing how much of his work he did not pirate.' " *Id.* (quoting *Sheldon v. Metro–Goldwyn Pictures Corp.,* 81 F.2d 49, 56 (2d Cir.1936)). In contrast, "the fact that a substantial portion of the infringing work was copied verbatim is evidence of the qualitative value of the copied material, both to the originator and to the plagiarist who seeks to profit from marketing someone else's copyrighted expression." *Id.* The Ninth Circuit has recognized that, while "wholesale copying does not preclude fair use per se, copying an entire work militates against a finding of fair use." *Worldwide Church of God,* 227 F.3d at 1118 (internal citation and quotation omitted). "If the secondary user only copies as much as is necessary for his or her intended use, then this factor will not weigh against him or her." *Kelly v.*

Arriba Soft Corp., 336 F.3d 811, 820–21 (9th Cir.2003). Under this factor, "attention turns to the persuasiveness of a parodist's justification for the particular copying done, and the enquiry will harken back to the first of the statutory factors ... [because] the extent of permissible copying varies with the purpose and character of the use." *Campbell*, 510 U.S. at 586–87, 114 S.Ct. 1164.

[56–58] The fourth and final factor focuses on "the effect of the use upon the potential market for or value of the copyrighted work." *Harper & Row*, 471 U.S. at 566, 105 S.Ct. 2218. This factor reflects the idea that fair use "is limited to copying by others which does not materially impair the marketability of the work which is copied." *Id.* at 566–67, 105 S.Ct. 2218. The Supreme Court has said that this factor is "undoubtedly the single most important element of fair use." *Id.* at 566, 105 S.Ct. 2218. It requires that courts "consider not only the extent of market harm caused by the particular actions of the alleged infringer, but also whether unrestricted and widespread conduct of the sort engaged in by the defendant ... would result in a substantially adverse impact on the potential market for the original." *Campbell*, 510 U.S. at 590, 114 S.Ct. 1164 (citation and quotation marks omitted). "Market harm is a matter of degree, and the importance of this factor will vary, not only with the amount of harm, but also with the relative strength of the showing on the other factors." *Id.* at 590 n. 21, 114 S.Ct. 1164.

Oracle asserts that all of these factors support its position that Google's use was not "fair use"—Google knowingly and illicitly copied a creative work to further its own commercial purposes, did so verbatim, and did so to the detriment of Oracle's market position. These undisputable facts, according to Oracle, should end the fair use inquiry. Oracle's position is not without force. On many of these points, Google does not debate Oracle's characterization of its conduct, nor could it on the record evidence.

Google contends, however, that, although it admittedly copied portions of the API packages and did so for what were purely commercial purposes, a reasonable juror still could find that: (1) Google's use was transformative; (2) the Java API packages are entitled only to weak protection; (3) Google's use was necessary to work within a language that had become an industry standard; and (4) the market impact on Oracle was not substantial.

[59] On balance, we find that due respect for the limit of our appellate function requires that we remand the fair use question for a new trial. First, although it is undisputed that Google's use of the API packages is commercial, the parties disagree on whether its use is "transformative." Google argues that it is, because it wrote its own implementing code, created its own virtual machine, and incorporated the packages into a smartphone platform. For its part, Oracle maintains that Google's use is not transformative because: (1) "[t]he same code in Android ... enables programmers to invoke the same pre-programmed functions in exactly the same way;" and (2) Google's use of the declaring code and packages does not serve a different function from Java. Appellant Reply Br. 47. While Google overstates what activities can be deemed transformative under a correct application of the law, we cannot say that there are no material facts in dispute on the question of whether Google's use is "transformative," even under a correct reading of the law. As such, we are unable to resolve this issue on appeal.

Next, while we have concluded that it was error for the trial court to focus *unduly* on the functional aspects of the packages, and on Google's competitive desire to

achieve commercial "interoperability" when deciding whether Oracle's API packages are entitled to copyright protection, we expressly noted that these factors may be relevant to a fair use analysis. While the trial court erred in concluding that these factors were sufficient to overcome Oracle's threshold claim of copyrightability, reasonable jurors might find that they are relevant to Google's fair use defense under the second and third factors of the inquiry. *See Sega*, 977 F.2d at 1524–25 (discussing the Second Circuit's approach to "break[ing] down a computer program into its component subroutines and subsubroutines and then identif[ying] the idea or core functional element of each" in the context of the second fair use factor: the nature of the copyrighted work). We find this particularly true with respect to those core packages which it seems may be necessary for anyone to copy if they are to write programs in the Java language. And, it may be that others of the packages were similarly essential components of any Java language-based program. So far, that type of filtration analysis has not occurred.

Finally, as to market impact, the district court found that "Sun and Oracle never successfully developed its own smartphone platform using Java technology." *Copyrightability Decision*, 872 F.Supp.2d at 978. But Oracle argues that, when Google copied the API packages, Oracle was licensing in the mobile and smartphone markets, and that Android's release substantially harmed those commercial opportunities as well as the potential market for a Java smartphone device. Because there are material facts in dispute on this factor as well, remand is necessary.

Ultimately, we conclude that this is not a case in which the record contains sufficient factual findings upon which we could base a de novo assessment of Google's affirmative defense of fair use. Accordingly, we remand this question to the district court for further proceedings. On remand, the district court should revisit and revise its jury instructions on fair use consistent with this opinion so as to provide the jury with a clear and appropriate picture of the fair use defense.[17]

II. Google's Cross-Appeal

Google cross-appeals from the portion of the district court's final judgment entered in favor of Oracle on its claim for copyright infringement as to the nine lines of rangeCheck code and the eight decompiled files. Final Judgment, *Oracle Am., Inc. v. Google Inc.*, No. 3:10–cv3561 (N.D.Cal. June 20, 2012), ECF No. 1211. Specifically, Google appeals from the district court's decisions: (1) granting Oracle's motion for

17. Google argues that, if we allow it to retry its fair use defense on remand, it is entitled to a retrial on infringement as well. We disagree. The question of whether Google's copying constituted infringement of a copyrighted work is ''distinct and separable'' from the question of whether Google can establish a fair use defense to its copying. *See Gasoline Prods. Co. v. Champlin Refining Co.*, 283 U.S. 494, 500, 51 S.Ct. 513, 75 L.Ed. 1188 (1931) (''Where the practice permits a partial new trial, it may not properly be resorted to unless it clearly appears that the issue to be retried is so distinct and separable from the others that a trial of it alone may be had without injustice.''). Indeed, we have emphasized more than once in this opinion the extent to which the questions are separable, and the confusion and error caused when they are blurred. The issues are not ''interwoven'' and it would not create ''confusion and uncertainty'' to reinstate the infringement verdict and submit fair use to a different jury. *Id.* We note, moreover, that, because Google only mentions this point in passing, with no development of an argument in support of it, under our case law, it has not been properly raised. *See SmithKline Beecham Corp. v. Apotex Corp.*, 439 F.3d 1312, 1320 (Fed.Cir.2006) (when a party provides no developed argument on a point, we treat that argument as waived) (collecting cases).

JMOL of infringement as to the eight decompiled Java files that Google copied into Android; and (2) denying Google's motion for JMOL with respect to rangeCheck.

[60–62] When reviewing a district court's grant or denial of a motion for JMOL, we apply the procedural law of the relevant regional circuit, here the Ninth Circuit. *Trading Techs. Int'l, Inc. v. eSpeed, Inc.*, 595 F.3d 1340, 1357 (Fed.Cir. 2010). The Ninth Circuit reviews a district court's JMOL decision de novo, applying the same standard as the district court. *Mangum v. Action Collection Serv., Inc.*, 575 F.3d 935, 938 (9th Cir. 2009). To grant judgment as a matter of law, the court must find that "the evidence presented at trial permits only one reasonable conclusion" and that "no reasonable juror could find in the non-moving party's favor." *Id.* at 938–39 (citation and internal quotation marks omitted).

Oracle explains that the eight decompiled files at issue "contain security functions governing access to network files" while rangeCheck "facilitates an important sorting function, frequently called upon during the operation of Java and Android." Oracle Response to Cross–Appeal 60–61. At trial, Google conceded that it copied the eight decompiled Java code files and the nine lines of code referred to as rangeCheck into Android. Its only defense was that the copying was de minimis. Accordingly, the district court instructed the jury that, "[w]ith respect to the infringement issues concerning the rangeCheck and other similar files, Google agrees that the accused lines of code and comments came from the copyrighted materials but contends that the amounts involved were so negligible as to be de minimis and thus should be excluded." Final Charge to the Jury (Phase One), *Oracle Am., Inc. v. Google, Inc.*, No. 3:10–cv–3561 (N.D.Cal. Apr. 30, 2012), ECF No. 1018, at 14.

Although the jury found that Google infringed Oracle's copyright in the nine lines of code comprising rangeCheck, it returned a noninfringement verdict as to eight decompiled security files. But because the trial testimony was that Google's use of the decompiled files was significant—and there was no testimony to the contrary—the district court concluded that "[n]o reasonable jury could find that this copying was de minimis." *Order Granting JMOL on Decompiled Files*, 2012 U.S. Dist. LEXIS 66417, at *6. As such, the court granted Oracle's motion for JMOL of infringement as to the decompiled security files.

On appeal, Google maintains that its copying of rangeCheck and the decompiled security files was de minimis and thus did not infringe any of Oracle's copyrights. According to Google, the district court should have denied Oracle's motion for JMOL "because substantial evidence supported the jury's verdict that Google's use of eight decompiled test files was de minimis." Cross–Appellant Br. 76. Google further argues that the court should have granted its motion for JMOL as to rangeCheck because the "trial evidence revealed that the nine lines of rangeCheck code were both quantitatively and qualitatively insignificant in relation to the [Java] platform." *Id.* at 78.

In response, Oracle argues that the Ninth Circuit does not recognize a de minimis defense to copyright infringement and that, even if it does, we should affirm the judgments of infringement on grounds that Google's copying was significant. Because we agree with Oracle on its second point, we need not address the first, except to note that there is some conflicting Ninth Circuit precedent on the question of whether there is a free-standing de minimis defense to copyright infringement or whether the substantiality of the alleged

copying is best addressed as part of a fair use defense. *Compare Norse v. Henry Holt & Co.*, 991 F.2d 563, 566 (9th Cir. 1993) (indicating that "even a small taking may sometimes be actionable" and the "question of whether a copying is substantial enough to be actionable may be best resolved through the fair use doctrine"), *with Newton v. Diamond*, 388 F.3d 1189, 1192–93 (9th Cir.2003) ("For an unauthorized use of a copyrighted work to be actionable, the use must be significant enough to constitute infringement. This means that even where the fact of copying is conceded, no legal consequences will follow from that fact unless the copying is substantial.") (internal citation omitted).[18]

[63] Even assuming that the Ninth Circuit recognizes a stand-alone de minimis defense to copyright infringement, however, we conclude that: (1) the jury reasonably found that Google's copying of the rangeCheck files was more than de minimis; and (2) the district court correctly concluded that the defense failed as a matter of law with respect to the decompiled security files.

First, the unrebutted testimony at trial revealed that rangeCheck and the decompiled security files were significant to both Oracle and Google. Oracle's expert, Dr. John Mitchell, testified that Android devices call the rangeCheck function 2,600 times just in powering on the device. Although Google argues that the eight decompiled files were insignificant because they were used only to test the Android platform, Dr. Mitchell testified that "using the copied files even as test files would have been significant use" and the district court specifically found that "[t]here was no testimony to the contrary." *Order Granting JMOL on Decompiled Files*, 2012 U.S. Dist. LEXIS 66417, at *6. Given this testimony, a reasonable jury could not have found Google's copying de minimis.

Google emphasizes that the nine lines of rangeCheck code "represented an infinitesimal percentage of the 2.8 million lines of code in the 166 Java packages—let alone the millions of lines of code in the entire [Java] platform." Google Cross–Appeal Br. 78–79. To the extent Google is arguing that a certain minimum number of lines of code must be copied before a court can find infringement, that argument is without merit. *See Baxter v. MCA, Inc.*, 812 F.2d 421, 425 (9th Cir.1987) ("[N]o bright line rule exists as to what quantum of similarity is permitted."). And, given the trial testimony that both rangeCheck and the decompiled security files are qualitatively significant and Google copied them in their entirety, Google cannot show that the district court erred in denying its motion for JMOL.

We have considered Google's remaining arguments and find them unpersuasive. Accordingly, we affirm both of the JMOL decisions at issue in Google's cross-appeal.

III. Google's Policy-Based Arguments

Many of Google's arguments, and those of some amici, appear premised on the belief that copyright is not the correct legal ground upon which to protect intellectual property rights to software programs; they opine that patent protection for such programs, with its insistence on

18. At least one recent district court decision has recognized uncertainty in Ninth Circuit law on this point. *See Brocade Commc'ns Sys. v. A10 Networks, Inc.*, No. 10–cv–3428, 2013 WL 831528, at *4 n. 3, 2013 U.S. Dist. LEXIS 8113, at *33 (N.D.Cal. Jan. 10, 2013) ("The Ninth Circuit has been unclear about whether the de minimis use doctrine serves as an affirmative defense under the Copyright Act's fair use exceptions or whether the doctrine merely highlights plaintiffs' obligation to show that 'the use must be significant enough to constitute infringement.' ") (citing *Newton*, 388 F.3d at 1193; *Norse*, 991 F.2d at 566).

non-obviousness, and shorter terms of protection, might be more applicable, and sufficient. Indeed, the district court's method of operation analysis seemed to say as much. *Copyrightability Decision*, 872 F.Supp.2d at 984 (stating that this case raises the question of "whether the copyright holder is more appropriately asserting an exclusive right to a functional system, process, or method of operation that belongs in the realm of patents, not copyrights"). Google argues that "[a]fter *Sega*, developers could no longer hope to protect [software] interfaces by copyright ... *Sega* signaled that the only reliable means for protecting the functional requirements for achieving interoperability was by patenting them." Appellee Br. 40 (quoting Pamela Samuelson, *Are Patents on Interfaces Impeding Interoperability* ? 93 Minn. L.Rev.1943, 1959 (2009)). And, Google relies heavily on articles written by Professor Pamela Samuelson, who has argued that "it would be best for a commission of computer program experts to draft a new form of intellectual property law for machine-readable programs." Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine–Readable Form*, 1984 Duke L.J. 663, 764 (1984). Professor Samuelson has more recently argued that "*Altai* and *Sega* contributed to the eventual shift away from claims of copyright in program interfaces and toward reliance on patent protection. Patent protection also became more plausible and attractive as the courts became more receptive to software patents." Samuelson, 93 Minn. L.Rev. at 1959.

Although Google, and the authority on which it relies, seem to suggest that software is or should be entitled to protection only under patent law—not copyright law—several commentators have recently argued the exact opposite. *See* Technology Quarterly, *Stalking Trolls*, ECONOMIST, Mar. 8, 2014, http://www.economist.com/news/technology-quarterly/21598 321–intellectual–property–after–being–blamed-stymying-innovation-america-vague ("[M]any innovators have argued that the electronics and software industries would flourish if companies trying to bring new technology (software innovations included) to market did not have to worry about being sued for infringing thousands of absurd patents at every turn. A perfectly adequate means of protecting and rewarding software developers for their ingenuity has existed for over 300 years. It is called copyright."); Timothy B. Lee, *Will the Supreme Court save us from software patents?*, WASH. POST, Feb. 26, 2014, 1:13 PM, http://www.washingtonpost.com/blogs/the-switch/wp/2014/02/26/will-the-supreme-court-save-us-from-software-patents/ ("If you write a book or a song, you can get copyright protection for it. If you invent a new pill or a better mousetrap, you can get a patent on it. But for the last two decades, software has had the distinction of being potentially eligible for both copyright and patent protection. Critics say that's a mistake. They argue that the complex and expensive patent system is a terrible fit for the fast-moving software industry. And they argue that patent protection is unnecessary because software innovators already have copyright protection available.").

Importantly for our purposes, the Supreme Court has made clear that "[n]either the Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted." *Mazer v. Stein*, 347 U.S. 201, 217, 74 S.Ct. 460, 98 L.Ed. 630 (1954). Indeed, the thrust of the CONTU Report is that copyright is "the most suitable mode of legal protection for computer software." Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan. L.Rev. 1045, 1072 (1989); *see also* CONTU Report at 1 (recommending that copyright law be amended "to make it explicit that

computer programs, to the extent that they embody an author's original creation, are proper subject matter of copyright"). Until either the Supreme Court or Congress tells us otherwise, we are bound to respect the Ninth Circuit's decision to afford software programs protection under the copyright laws. We thus decline any invitation to declare that protection of software programs should be the domain of patent law, and only patent law.

Conclusion

For the foregoing reasons, we conclude that the declaring code and the structure, sequence, and organization of the 37 Java API packages at issue are entitled to copyright protection. We therefore reverse the district court's copyrightability determination with instructions to reinstate the jury's infringement verdict. Because the jury hung on fair use, we remand Google's fair use defense for further proceedings consistent with this decision.

With respect to Google's cross-appeal, we affirm the district court's decisions: (1) granting Oracle's motion for JMOL as to the eight decompiled Java files that Google copied into Android; and (2) denying Google's motion for JMOL with respect to the rangeCheck function. Accordingly, we affirm-in-part, reverse-in-part, and remand for further proceedings.

AFFIRMED–IN–PART, REVERSED–IN–PART, AND REMANDED

886 F.3d 1179
United States Court of Appeals, Federal Circuit.

ORACLE AMERICA, INC., Plaintiff–Appellant
v.
GOOGLE LLC, Defendant–Cross–Appellant

2017-1118, 2017-1202
|
Decided: March 27, 2018

Synopsis

Background: Developer of computer programming platform brought action alleging that technology company's unauthorized use of its application programming interface (API) packages to create mobile smartphone operating system infringed its copyrights. After jury entered verdict in company's favor, the United States District Court for the Northern District of California, No. 3:10-cv-03561-WHA, William H. Alsup, J., denied developer's motion for judgment as matter of law (JMOL), 2016 WL 3181206, and its renewed motion for JMOL and for new trial, 2016 WL 5393938. Developer appealed.

Holdings: The Court of Appeals, O'Malley, Circuit Judge, held that:

[1] company's use of API packages was commercial in nature;

[2] company's use of API packages was not "transformative";

[3] developer suffered actual and potential harm as result of company's copying of its API packages; and

[4] company's use of API packages was not fair use.

Reversed and remanded.

***1183** Appeals from the United States District Court for the Northern District of California in No. 3:10-cv-03561-WHA, Judge William H. Alsup.

Attorneys and Law Firms

E. Joshua Rosenkranz, Orrick, Herrington & Sutcliffe LLP, New York, NY, argued for plaintiff-appellant. Also represented by Peter A. Bicks, Matthew Lee Bush, Andrew D. Silverman, Lisa Simpson; Melanie L. Bostwick, Kelsi Brown Corkran, Mark S. Davies, Jeremy Peterman, Washington, DC; Annette Louise Hurst, San Francisco, CA; Dale M. Cendali, Joshua L. Simmons, Kirkland & Ellis LLP, New York, NY; Ruchika Agrawal, Dorian Estelle Daley, Deborah Kay Miller, Matthew Sarboraria, Oracle America, Inc., Redwood Shores, CA.

Daryl Joseffer, King & Spalding LLP, Washington, DC, argued for defendant-cross-appellant. Also represented by Bruce William Baber, Atlanta, GA; Christa M. Anderson, Steven A. Hirsch, Michael Soonuk Kwun, Reid P. Mullen, Robert A. Van Nest, Keker, Van Nest & Peters LLP, San Francisco, CA; Renny F. Hwang, Google LLC, Mountain View, CA.

Kenneth L. Doroshow, Jenner & Block LLP, Washington, DC, for amicus curiae The Copyright Alliance. Also represented by Erica Lauren Ross.

Daniel J. Brooks, Scarola Malone & Zubatov LLP, New York, NY, for amicus curiae New York Intellectual Property Law Association. Also represented by Charles R. Macedo, Amster Rothstein & Ebenstein LLP, New York, NY; Annemarie Hassett, NYU School of Law, New York, NY.

Jared Bobrow, Weil, Gotshal & Manges LLP, Redwood Shores, CA, for amici curiae Eugene H. Spafford, Zhi Ding, Adam Porter, Ken Castleman. Also represented by Amanda Branch, Aaron Y. Huang.

Steven Thomas Cottreau, Clifford Chance Rogers & Wells LLP, Washington, DC, for amici curiae Scott McNealy, Brian Sutphin.

Marc Robert Lewis, Lewis & Llewellyn LLP, San Francisco, CA, for amicus curiae Ralph Oman. Also represented by Evangeline Zimmerman Burbidge.

Robert H. Rotstein, Mitchell, Silberberg & Knupp, LLP, Los Angeles, CA, for amici curiae Motion Picture Association of America, Inc., Independent Film & Television Alliance. Also represented by J. Matthew Williams, Washington, DC.

Duncan W. Crabtree–Ireland, SAG–AFTRA, Los Angeles, CA, for amicus curiae Screen Actors Guild—American Federation of Television and Radio Artists.

William M. Jay, Goodwin Procter LLP, Washington, DC, for amici curiae Recording Industry Association of America, Association of American Publishers. Also represented by Andrew Kim. Recording Industry Association of America also represented by George M. Borkowski, Mitchell, Silberberg & Knupp, LLP, Los Angeles, CA.

Rebecca Murphy Thompson, Competitive Carriers Association, Washington, DC, for amicus curiae Competitive Carriers Association.

Antigone Gabriella Peyton, Protorae Law PLLC, Tysons, VA, for amici curiae Sandra Aistars, Matthew Barblan, Jon A. Baumgarten, Stephen Carlisle, Jon M. Garon, Hugh Hansen, Devlin Hartline, Jiarui Liu, Adam Mossoff, Raymond T. Nimmer, Eric Priest, Sean M. O'Connor, Mark F. Schultz.

Lindsay Warren Bowen, Cowan, DeBaets, Abrahams & Sheppard LLP, New York, NY, for amici curiae PACA, Digital Media Licensing Association, Inc., Graphic Artists Guild, National Press Photographers Association, North American Nature Photography Association, American Society of Media Photographers, Inc., American Photographic Artists, Professional Photographers of America. Also represented by Scott J. Sholder.

Richard L. Rainey, Covington & Burling LLP, Washington, DC, for amicus curiae BSA | The Software Alliance. Also represented by Peter Andrew Swanson.

Jonathan Band, Jonathan Band PLLC, Washington, DC, for amicus curiae Computer & Communications Industry Association. Also represented by Matthew Schruers, Computer & Communications Industry Association, Washington, DC.

Michael Barclay, Electronic Frontier Foundation, San Francisco, CA, for amici curiae Electronic Frontier Foundation, Public Knowledge. Also represented by Mitchell L. Stoltz.

Marcia Hofmann, Zeitgeist Law PC, San Francisco, CA, for amicus curiae Mozilla Corporation.

Richard M. Brunell, American Antitrust Institute, Washington, DC, for amicus curiae American Antitrust Institute. Also represented by Shubha Ghosh, Syracuse University College of Law, Syracuse, NY.

Jeffrey A. Lamken, MoloLamken LLP, Washington, DC, for amici curiae Microsoft Corp., Red Hat, Inc., Hewlett Packard Enterprise Company. Also represented by Rayiner Hashem, Michael Gregory Pattillo, Jr.; Lisa Wang Bohl, Chicago, IL.

Jason Michael Schultz, NYU School of Law, New York, NY, for amici curiae Timothy K. Armstrong, Cark D. Asay, Shyamkrishna Balganesh, Ann Bartow, Oren Bracha, Annemarie Bridy, Dan L. Burk, Michael A. Carrier, Michael W. Carroll, Andrew Chin, Julie E. Cohen, Kevin Collins, Rebecca Curtin, Ben Depoorter, Roger Allan Ford, Brian L. Frye, Jim Gibson, Eric Goldman, James Grimmelmann, Peter Jaszi, Yvette Joy Liebesman, Jessica Litman, Brian J. Love, Michael J. Madison, Mark. P. McKenna, Joseph Scott Miller, Deirdre K. Mulligan, Tyler T. Ochoa, Aaron Perzanowski, Victoria F. Phillips, Arti K. Rai, Jerome H. Reichman, Michael Rustad, Matthew Sag, Pamela Samuelson, Jessica Sibley, Joshua David Sarnoff, Lea Shaver, Christopher Jon Sprigman, Katherine J. Strandburg, Rebecca Tushnet, Jennifer M. Urban.

Phillip R. Malone, Stanford Law School, Stanford, CA, for amici curiae Harold Abelson, Tom Ball, Brian Behlendorf, Gordon Bell, Jon Bentley, Matthew Bishop, Joshua Bloch, Dan Boneh, Gilad Bracha, Eric Brewer, Frederick Brooks, Rick Cattell, Vinton G. Cerf, William Cook, Mark Davis, Miguel de Icaza, Jeffrey Dean, L. Peter Deutsch, Whitfield Diffie, David L. Dill, Lester Earnest, Brendan Eich, Dawson Engler, Martin Fowler, Neal Gafter, Robert Harper, John Hennessy, Tom Jennings, Alan Kay, Brian Kernighan, David Klausner, Ray Kurzweil, Kin Lane, Ed Lazowska, Doug Lea, Bob Lee, Sheng Liang, Barbara Liskov, Paul Menchini, Andrew W. Moore, James H. Morris, Peter Norvig, Martin Odersky, Tim Paterson, David Patterson, Alex Payne, Tim Peierls, Simon Phipps, Bill Pugh, Ronald L. Rivest, Curtis Schroeder, Robert Sedgewick, Mary Shaw, Barbara Simons, Dave Snigier, Alfred Z. Spector, Bjarne Stroustrup, Gerald Jay Sussman, Ivan E. Sutherland, Andrew Tanenbaum, Brad Templeton, Ken Thompson, Michael Tiemann, Linus Torvalds, Andrew Tridgell, Jeffrey Ullman, Andries Van Dam, Guido Van Rossum,

John Villasenor, Jan Vitek, Philip Wadler, James H. Waldo, Daniel S. Wallach, Peter J. Weinberger, Steve Wozniak, Frank Yellin. Also represented by Jeffrey Theodore Pearlman.

Mark A. Lemley, Durie Tangri LLP, San Francisco, CA, for amici curiae Engine Advocacy, The App Developers Alliance, GitHub, Inc. Also represented by Joseph Gratz, Clement Roberts.

Before O'Malley, Plager, and Taranto, Circuit Judges.

Opinion

O'Malley, Circuit Judge.

***1184** ***1185** This copyright case returns to us after a second jury trial, this one focusing on the defense of fair use. Oracle America, Inc. ("Oracle") filed suit against Google Inc. ("Google") [1] in the United States District Court for the Northern District of California, alleging that Google's unauthorized use of 37 packages of Oracle's Java application programming interface ("API packages") in its Android operating system infringed Oracle's patents and copyrights.

At the first trial, the jury found that Google infringed Oracle's copyrights in the Java Standard Edition platform, but deadlocked on the question of whether Google's copying was a fair use. [2] After the verdict, however, the district court found that the API packages were not copyrightable as a matter of law and entered judgment for Google. *Oracle Am., Inc. v. Google Inc.*, 872 F.Supp.2d 974 (N.D. Cal. 2012). Oracle appealed that determination to this court, and we reversed, finding that declaring code and the structure, sequence, and organization ("SSO") of the Java API packages are entitled to copyright protection. *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1348 (Fed. Cir. 2014). We remanded with instructions to reinstate the jury's infringement verdict and for further proceedings on Google's fair use defense and, if appropriate, on damages. *Id.* at 1381.

Google subsequently filed a petition for certiorari on the copyrightability determination. The Supreme Court called for the views of the Solicitor General, who expressed agreement with our determination and recommended denying review. The Supreme Court denied certiorari in 2015. *Google Inc. v. Oracle Am., Inc.*, —— U.S. ——, 135 S.Ct. 2887, 192 L.Ed.2d 948 (2015) (Mem.).

At the second jury trial, Google prevailed on its fair use defense. After the jury verdict, the district court denied Oracle's motion for judgment as a matter of law ("JMOL") and entered final judgment in favor of Google. *Oracle Am., Inc. v. Google Inc.*, No. C 10-03561, 2016 WL 3181206 (N.D. Cal. June 8, 2016) ("*Order Denying JMOL*"); Final Judgment, ***1186** *Oracle Am., Inc. v. Google Inc.*, No. 3:10-cv-3561 (N.D. Cal. June 8, 2016), ECF No. 1989. Oracle filed a renewed motion for JMOL and separately moved for a new trial. The district court denied both motions in a single order. *Oracle Am., Inc. v. Google Inc.*, No. C 10-03561, 2016 WL 5393938 (N.D. Cal. Sept. 27, 2016) ("*Order Denying Renewed JMOL/New Trial*"). Consistent with these determinations, no damages verdict was rendered.

Oracle now appeals from the district court's final judgment and its decisions denying Oracle's motions for JMOL and motion for a new trial. Google cross-appeals from the final judgment purportedly to "preserv[e] its claim that the declarations/SSO are not protected by copyright law," but advances no argument for why this court can or should revisit our prior decision on copyrightability. Cross–Appellant Br. 83.

Because we conclude that Google's use of the Java API packages was not fair as a matter of law, we reverse the district court's decisions denying Oracle's motions for JMOL and remand for a trial on damages. We also dismiss Google's cross-appeal.

I. BACKGROUND

A. The Technology

Oracle's predecessor, Sun Microsystems, Inc. ("Sun"), developed the Java platform for computer programming in the 1990s, and Oracle purchased Sun in 2010. The Java platform is software used to write and run programs in the Java programming language. It allows programmers to write programs that "run on different types of computer hardware without having to rewrite them for each different type." *Oracle*, 750 F.3d at 1348. With Java, programmers can "write once, run anywhere." *Id.*

The Java 2 Standard Edition ("Java SE") of the platform includes, among other things, the Java Virtual Machine

and the Java Application Programming Interface ("API"). The Java API is a collection of "pre-written Java source code programs for common and more advanced computer functions." *Order Denying JMOL*, 2016 WL 3181206, at *3. These APIs "allow programmers to use the prewritten code to build certain functions into their own programs rather than write their own code to perform those functions from scratch. They are shortcuts." *Oracle*, 750 F.3d at 1349. The prewritten programs are organized into packages, classes, and methods. Specifically, an API package is a collection of classes and each class contains methods and other elements. "Each method performs a specific function, sparing a programmer the need to write Java code from scratch to perform that function." *Order Denying JMOL*, 2016 WL 3181206, at *3.

To include a particular function in a program, the programmer invokes the Java "declaring code." As the district court explained, the declaring code is the line or lines of source code that "declares or defines (i) the method name and (ii) the input(s) and their type as expected by the method and the type of any outputs." *Id.* at *4. After the declaring code, each method includes "implementing code," which takes the input(s) and gives the computer step-by-step instructions to carry out the declared function.

By 2008, Java SE included 166 API packages divided into 3,000 classes containing more than 30,000 methods. At issue in this appeal are 37 API packages from Java SE Version 1.4 and Version 5.0. We have already concluded that the declaring code and the SSO of the 37 Java API packages at issue are entitled to copyright protection. *Oracle*, 750 F.3d at 1348.

***1187** The Java programming language itself is free and available for use without permission. At this stage, it is undisputed that, to write in the Java programming language, "62 classes (and some of their methods), spread across three packages *within* the Java API library, *must* be used. Otherwise the language itself will fail." *Order Denying JMOL*, 2016 WL 3181206, at *5. It is also undisputed that anyone using the Java programming language can write their own library of prewritten programs to carry out various functions.

Although Oracle makes the Java platform freely available to programmers building applications ("apps"), it devised a licensing scheme to attract programmers while simultaneously commercializing the platform. In relevant part, Oracle charges a licensing fee to those who want to use the APIs in a competing platform or embed them in an electronic device. To preserve the "write once, run anywhere" philosophy, Oracle imposes strict compatibility requirements on licensees. *Oracle*, 750 F.3d at 1350. Oracle also made available without charge under an open source license a version of Java called "OpenJDK." *Order Denying JMOL*, 2016 WL 3181206, at *10. Oracle maintains, however, that OpenJDK came with an important catch: any company that improved on the packages in OpenJDK had to " 'give away those changes for free' to the Java community." Appellant Br. 53.

The evidence showed that Oracle licensed Java in 700 million PCs by 2005. Although Oracle never successfully developed its own smartphone platform using Java, it licensed Java SE for mobile devices. According to Oracle, the "mobile device market was particularly lucrative," and "Java quickly became the leading platform for developing and running apps on mobile phones." Appellant Br. 9.

B. Google's Android Platform

In 2005, Google acquired Android, Inc. as part of a plan to develop a software platform for mobile devices. That same year, Google and Sun began discussing the possibility of Google taking a license to use and adapt the Java platform for mobile devices. *Oracle*, 750 F.3d at 1350. The parties were unable to reach an agreement, in part because Google wanted device manufacturers to be able to use Oracle's APIs in Android for free with no limits on modifying the code, which would jeopardize the "write once, run anywhere" philosophy.

The jury heard evidence that Google wanted to move quickly to develop a platform that would attract Java developers to build apps for Android. The Android team had been working on creating its own APIs, but was unable to do so successfully. After negotiations between the parties reached an impasse, Google elected to "[d]o Java anyway and defend [its] decision, perhaps making enemies along the way." *Order Denying JMOL*, 2016 WL 3181206, at *6. It is undisputed that Google copied verbatim the declaring code of the 37 Java API packages —11,500 lines of Oracle's copyrighted code. It also copied the SSO of the Java API packages. Google then wrote its own implementing code.

Google announced its Android software platform for mobile devices in 2007, and the first Android phones went on sale the following year. Google provides the Android platform free of charge to smartphone manufacturers and publishes the source code for use without charge under an open source license. Although Google does not directly charge its users, Android has generated over $42 billion in revenue from advertising. Oracle explains that Android was "devastating" to its licensing strategy and that many of its customers switched to Android. Appellant Br. 15. ***1188** Even customers who stayed with Oracle cited Android as a reason to demand discounts. The jury heard evidence that Amazon, which had entered into a license to use Java for its Kindle tablet device, switched to Android for the subsequently released Kindle Fire and then used the existence of Android to leverage a steep discount from Oracle on the next generation Kindle.

C. Remand Proceedings

In the first appeal, we held that the declaring code and the SSO of the 37 API packages are entitled to copyright protection and ordered the district court to reinstate the jury's infringement finding. *Oracle*, 750 F.3d at 1381. We also considered Oracle's argument that it was entitled to judgment as a matter of law on Google's fair use defense. Although we found that Oracle's position was "not without force," and that Google was overstating what could be fair use under the law, we found that the record evidence regarding the relevant fair use factors was insufficiently developed for us to resolve the issue on appeal. *Oracle*, 750 F.3d at 1376. In doing so, we pointed to sharp disputes between the parties, both legal and factual, including whether Google's use was transformative, whether "functional aspects of the package" and Google's "desire to achieve commercial 'interoperability' " weighed in favor of the second and third factors, and whether Android caused market harm to Oracle. *Id.* at 1376–77. We concluded that "due respect for the limit of our appellate function" required remand. *Id.* at 1376.

During the pendency of the first appeal, Google's Android business expanded significantly. Android gained new users and developers, and Google "released modified implementations and derivatives of Android for use in numerous device categories, including wearable devices with small screens (Android Wear), dashboard interfaces in cars (Android Auto), television sets (Android TV), and everyday devices with Internet connectivity." *Oracle Am., Inc. v. Google Inc.*, No. C10-03561, 2016 WL 1743111, at *1 (N.D. Cal. May 2, 2016) ("*Order on Motion in Limine*").

When the case returned to the district court, Oracle filed a supplemental complaint adding allegations of market harm and damages resulting from new versions of Android released since the original complaint. Specifically, Oracle alleged that Google had launched new versions of Android for phones and tablets and had expanded Android into new device categories. *Id.* Google did not oppose the supplemental complaint, and the district court granted Oracle's motion to file it. But when Oracle served expert reports that addressed versions of Java SE that were not at issue in the first trial, Google moved to strike those reports. *Id.*

When the parties were unable to agree on the scope of the retrial, the district court limited it to: (1) the two versions of Java SE that Oracle asserted in the first trial; and (2) released versions of Android used in smartphones and tablets "which Google ... agreed would be subject to the prior jury's adverse finding of infringement and which Oracle identified in its supplemental complaint." *Id.* The court explained that Oracle retained the right to sue Google for infringement with respect to the other versions and implementations of Android in a separate trial or proceeding. Order re: Google's Motion to Strike at 2, *Oracle Am., Inc. v. Google Inc.*, No. 3:10-cv-3561 (N.D. Cal. Feb. 5, 2016), ECF No. 1479. The court also granted Google's motion in limine to exclude all evidence of the new Android products.

The district court bifurcated the issue of fair use from willfulness and monetary remedies, and the trial on fair use began ***1189** on May 10, 2016. After roughly one week of evidence and several days of deliberations, the jury found that Google's use of the declaring lines of code and the SSO of the 37 API packages constituted fair use.

Oracle moved for JMOL, which the district court denied. At the outset, the court noted that Oracle stipulated before the jury "that it was fair to use the 62 'necessary' classes given that the Java programming language itself was free and open to use without a license." *Order Denying JMOL*, 2016 WL 3181206, at *5. "That the 62 'necessary' classes reside without any identification as such within

the Java API library (rather than reside within the programming language)," the court explained, "supports Google's contention that the Java API library is simply an extension of the programming language itself and helps explain why some view the Java API declarations as free and open for use as the programming language itself." *Id.* Because Android and Java both "presupposed the Java programming language in the first place," the court noted that a jury reasonably could have found that it "was better for both to share the same SSO insofar as they offered the same functionalities, thus maintaining usage consistency across systems and avoiding cross-system confusion." *Id.* at *6.

The district court then considered each of the four statutory fair use factors. As to factor one—the purpose and character of the use—the court concluded that a reasonable jury could have found that, although Google's use was commercial, it was transformative because Google integrated only selected elements for mobile smartphones and added its own implementing code. *Id.* at *7–9. With respect to factor two—the nature of the copyrighted work—the district court found that a reasonable jury could have concluded that, "while the declaring code and SSO were creative enough to qualify for copyright protection," they were not "highly creative," and that "functional considerations predominated in their design." *Id.* at *10.

As to factor three—the amount and substantiality of the portion used—the court concluded that a reasonable jury could have found that "Google copied only so much as was reasonably necessary for a transformative use," and that the number of lines duplicated was minimal. *Id.* Finally, as to factor four—market harm—the court concluded 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." *Id.* The court determined that, on the record presented, the jury could have found for either side and that the jury was "reasonably within the record in finding fair use." *Id.* at *11.

Oracle subsequently renewed its motion for JMOL and separately moved for a new trial challenging several of the court's discretionary decisions at trial. The district court denied both motions in a single order. With respect to JMOL, the court simply stated that it denied Oracle's renewed motion for the same reasons it denied the original motion. With respect to the motion for a new trial, the court rejected Oracle's argument that the court abused its discretion by limiting the evidence at trial to Google's use of Android in smartphones and tablets.

The court also rejected Oracle's allegation that Google engaged in discovery misconduct by withholding evidence during discovery relating to Google's App Runtime for Chrome ("ARC"), which enabled laptops and desktops running Google's computer operating system to run certain Android applications. *Order Denying Renewed JMOL/New Trial*, ***1190** 2016 WL 5393938, at *5. The court found that Google had produced relevant documents during discovery and that, in any event, those documents pertained to issues beyond the scope of the retrial. *Id.* at *7–8.

Finally, the district court rejected Oracle's argument that certain of the court's evidentiary rulings were abuses of discretion. The court explained that it: (1) redacted one line from an email because it was "too inflammatory and without foundation;" and (2) excluded other documents because Oracle had withheld them as privileged until trial. *Id.* at *9–12.

On June 8, 2016, the district court entered final judgment in favor of Google and against Oracle. Oracle timely appealed from the district court's judgment against it, including the court's underlying decisions denying its motions for JMOL and for a new trial. Google timely cross-appealed from all adverse orders and rulings underlying that final judgment.

[1] **[2]** This court has exclusive jurisdiction over all appeals in actions involving patent claims, including where, as here, an appeal raises only non-patent issues. 28 U.S.C. § 1295(a)(1). Because copyright law is not within this court's exclusive jurisdiction, we apply the law of the regional circuit in which the district court sits; here, the Ninth Circuit. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 837 (Fed. Cir. 1992).

II. ORACLE'S APPEAL

A. Legal Framework

It is undisputed that Google copied Oracle's declaring code and SSO for the 37 API packages verbatim. The

question is whether that copying was fair. "From the infancy of copyright protection, some opportunity for fair use of copyrighted materials has been thought necessary to fulfill copyright's very purpose, 'to promote the Progress of Science and useful Arts.' " *Campbell v. Acuff–Rose Music, Inc.*, 510 U.S. 569, 575, 114 S.Ct. 1164, 127 L.Ed.2d 500 (1994) (quoting U.S. Const., art. I, § 8, cl. 8). As the Supreme Court noted in *Campbell*, "[i]n truth, in literature, in science and in art, there are, and can be, few, if any, things, which in an abstract sense, are strictly new and original throughout. Every book in literature, science and art, borrows, and must necessarily borrow, and use much which was well known and used before." *Id.* (quoting *Emerson v. Davies*, 8 F. Cas. 615, 619 (C.C.D. Mass. 1845)).

The fair use defense began as a judge-made doctrine and was codified in Section 107 of the 1976 Copyright Act. *Id.* at 576, 114 S.Ct. 1164. It operates as a limited exception to the copyright holder's exclusive rights and permits use of copyrighted work if it is "for purposes such as criticism, comment, news reporting, teaching ..., scholarship, or research." 17 U.S.C. § 107. The "such as" language confirms that the listing "was not intended to be exhaustive," but nevertheless "give[s] some idea of the sort of activities the courts might regard as fair use under the circumstances." *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 561, 105 S.Ct. 2218, 85 L.Ed.2d 588 (1985) (citation omitted).

"Section 107 requires a case-by-case determination whether a particular use is fair, and the statute notes four nonexclusive factors to be considered." *Id.* at 549, 105 S.Ct. 2218. Those factors include: (1) "the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;" (2) "the nature of the copyrighted work;" (3) "the amount and substantiality of the portion used in relation to the copyrighted work as a whole;" and (4) "the effect of the use upon the potential market for or value of the copyrighted ***1191** work." 17 U.S.C. § 107. The Supreme Court has cautioned against adopting bright-line rules and has emphasized that all of the statutory factors "are to be explored, and the results weighed together, in light of the purposes of copyright." *Campbell*, 510 U.S. at 578, 114 S.Ct. 1164.

[3] The legislative history reveals that Congress intended § 107 " 'to restate the present judicial doctrine of fair use, not to change, narrow, or enlarge it in any way' and intended that courts continue the common-law tradition of fair use adjudication." *Id.* at 577, 114 S.Ct. 1164 (quoting H.R. Rep. No. 94-1476, at 66 (1976), S. Rep. No. 94-473 at 62 (1975), U.S. Code Cong. & Admin. News 5659, 5679 (1976)). Accordingly, in balancing the four statutory factors, courts consider "whether the copyright law's goal of 'promot[ing] the Progress of Science and useful Arts,' U.S. Const., art. 1, § 8, cl. 8, 'would be better served by allowing the use than by preventing it.' " *Castle Rock Entm't, Inc. v. Carol Publ'g Grp., Inc.*, 150 F.3d 132, 141 (2d Cir. 1998) (quoting *Arica Inst., Inc. v. Palmer*, 970 F.2d 1067, 1077 (2d Cir. 1992)).

[4] Despite this guidance, the doctrine of fair use has long been considered "the most troublesome in the whole law of copyright." *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)). It both permits and requires "courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster." *Campbell*, 510 U.S. at 577, 114 S.Ct. 1164 (quoting *Stewart v. Abend*, 495 U.S. 207, 236, 110 S.Ct. 1750, 109 L.Ed.2d 184 (1990)).

[5] **[6]** Because fair use is an affirmative defense to a claim of infringement, Google bears the burden to prove that the statutory factors weigh in its favor. *Id.* at 590, 114 S.Ct. 1164. Not all of the four factors must favor Google, however. *See Wall Data Inc. v. L.A. Cty. Sheriff's Dep't*, 447 F.3d 769, 778 (9th Cir. 2006). Instead, "fair use is appropriate where a 'reasonable copyright owner' would have consented to the use, i.e., where the 'custom or public policy' at the time would have defined the use as reasonable." *Id.* (citation omitted).

On appeal, Oracle argues that each of the four statutory factors weighs against a finding of fair use. Specifically, it submits that: (1) the purpose and character of Google's use was purely for commercial purposes; (2) the nature of Oracle's work is highly creative; (3) Google copied 11,330 more lines of code than necessary to write in a Java language-based program; and (4) Oracle's customers stopped licensing Java SE and switched to Android because Google provided free access to it. In the alternative, Oracle argues that it is entitled to a new trial because the district court made several errors that deprived it of a fair opportunity to present its case.

Because, as explained below, we agree with Oracle that Google's copying was not fair use as a matter of law, we need not address Oracle's alternative arguments for a new trial.

B. Standards of Review

Before turning to a consideration of the four statutory factors and any relevant underlying factual determinations, we first address the standard of review we are to employ in that consideration. While this section of most appellate opinions presents easily resolvable questions, like much else in the fair use context, that is not completely the case here.

There are several components to this inquiry. First, which aspects of the fair use determination are legal in nature and which are factual? Particularly, is the ultimate ***1192** question of fair use a legal inquiry which is to be reviewed de novo? Second, what factual questions are involved in the fair use determination and under what standard are those determinations to be reviewed? Finally, though neither party addresses the question in detail, we consider what, if any, aspects of the fair use determination are for the jury to decide.

The Supreme Court has said that fair use is a mixed question of law and fact. *Harper & Row*, 471 U.S. at 560, 105 S.Ct. 2218 (citing *Pac. & S. Co. v. Duncan*, 744 F.2d 1490, 1495 n.8 (11th Cir. 1984)). Merely characterizing an issue as a mixed question of law and fact does not dictate the applicable standard of review, however. *See U.S. Bank Nat'l Ass'n ex rel. CWCapital Asset Mgmt. LLC*, —— U.S. ——, 138 S.Ct. 960, —— L.Ed.2d ——, 2018 WL 1143822, at *5 (2018).

The Supreme Court has recently explained how we are to determine what the standard of review should be in connection with any mixed question of law and fact. *Id.* Specifically, the Court made clear that an appellate court is to break mixed questions into their component parts and to review each under the appropriate standard of review. *Id.* at *5–7. In *U.S. Bank*, the Supreme Court considered the level of review to be applied to a Bankruptcy Court's determination of whether a creditor in a bankruptcy action qualified as a "non-statutory insider" for purposes of 11 U.S.C. § 1129(a). *Id.* at *3–4. The Court found that there were three components to that inquiry: (1) determining the legal standard governing the question posed and what types of historical facts are relevant to that standard; (2) finding what the historical facts in the case at hand are; and (3) assessing whether the historical facts found satisfy the legal test governing the question to be answered. *Id.* at *4–5. As the Court explained, the first of these three is a purely legal question to be reviewed de novo on appeal and the second involves factual questions which "are reviewable only for clear error." *Id.* at *4 (citing Fed. R. Civ. P. 52(a)(6) (clear error standard)). The third is what the Court characterized as the "mixed question." *Id.* at *5.

[7] Importantly, the Court noted that "[m]ixed questions are not all alike." *Id.* The Court then held that "the standard of review for a mixed question all depends—on whether answering it entails primarily legal or factual work." *Id.* Where applying the law to the historical facts "involves developing auxiliary legal principles of use in other cases—appellate courts should typically review a decision de novo." *Id.* (citing *Salve Regina College v. Russell*, 499 U.S. 225, 231–33, 111 S.Ct. 1217, 113 L.Ed.2d 190 (1991)). But where the mixed question requires immersion in case-specific factual issues that are so narrow as to "utterly resist generalization," the mixed question review is to be deferential. *Id.* (quoting *Pierce v. Underwood*, 487 U.S. 552, 561–62, 108 S.Ct. 2541, 101 L.Ed.2d 490 (1988)). Ultimately, the Court found that review of the mixed question at issue in that bankruptcy context should be deferential because de novo review of the question would do little to "clarify legal principles or provide guidance to other courts resolving other disputes." *Id.* at *7.

While this may be the first time the Supreme Court has so clearly explained how appellate courts are to analyze mixed questions of law and fact, it is not the first time the Supreme Court has told us how to analyze the particular mixed question of law and fact at issue here. In other words, while the Supreme Court has not previously broken the fair use inquiry into its three analytical components as expressly as it did the question in *U.S. Bank*, it has made ***1193** clear that both the first and third of those components are subject to de novo review.

In *Harper & Row*, the Court explained that, "[w]here the district court has found facts sufficient to evaluate each of the statutory factors, an appellate court 'need not remand for further factfinding but may conclude as a matter of

law that the challenged use does not qualify as a fair use of the copyrighted work.' " 471 U.S. at 560, 105 S.Ct. 2218 (quoting *Pac. & S. Co.*, 744 F.2d at 1495) (internal alterations omitted)). The Ninth Circuit has resolved the question in the same way. Where fair use is resolved on summary judgment, the Ninth Circuit reviews the district court's ultimate determination de novo. *SOFA Entm't, Inc. v. Dodger Prods., Inc.*, 709 F.3d 1273, 1277 (9th Cir. 2013) ("Whether Dodger's use of the clip constitutes fair use is a mixed question of law and fact that we review de novo."). That court has explained that, " 'as fair use is a mixed question of fact and law, so long as the record is "sufficient to evaluate each of the statutory factors," we may reweigh on appeal the inferences to be drawn from that record.' " *Mattel, Inc. v. Walking Mountain Prods.*, 353 F.3d 792, 800 (9th Cir. 2003) (quoting *L.A. News Serv. v. CBS Broad., Inc.*, 305 F.3d 924, 942 (9th Cir. 2002)).

This treatment of the ultimate question posed when a fair use defense is raised makes sense. The fair use question entails, in the words of *U.S. Bank*, a primarily legal exercise. It requires a court to assess the inferences to be drawn from the historical facts found in light of the legal standards outlined in the statute and relevant case law and to determine what conclusion those inferences dictate. Because, as noted below, the historical facts in a fair use inquiry are generally few, generally similar from case to case, and rarely debated, resolution of what any set of facts means to the fair use determination definitely does not "resist generalization." *See U.S. Bank*, 138 S.Ct. at 966. Instead, the exercise of assessing whether a use is fair in one case will help guide resolution of that question in all future cases.

[8] For these reasons, we conclude that whether the court applied the correct legal standard to the fair use inquiry is a question we review de novo, whether the findings relating to any relevant historical facts were correct are questions which we review with deference, and whether the use at issue is ultimately a fair one is something we also review de novo.

We have outlined the legal standard governing fair use above. We consider below whether the court properly applied those standards in the course of its fair use analysis and whether it reached the correct legal conclusion with respect to fair use. Before doing so, we briefly discuss the historical facts relevant to the fair use inquiry and consider the jury's role in determining those facts.

[9] The Supreme Court has described "historical facts" as "a recital of external events." *Thompson v. Keohane*, 516 U.S. 99, 110, 116 S.Ct. 457, 133 L.Ed.2d 383 (1995); *see also U.S. Bank*, 138 S.Ct. at 965 (describing the historical facts at issue there as facts relating to "the attributes of a particular relationship or the circumstances and terms of a prior transaction"). In the fair use context, historical facts include the "origin, history, content, and defendant's use" of the copyrighted work. *Fitzgerald v. CBS Broad., Inc.*, 491 F.Supp.2d 177, 184 (D. Mass. 2007); *see also Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F.Supp. 78, 95 (D. Mass 1992) (defining historical facts to include "who did what, where, and when"). When asked at oral argument to identify historical facts relevant to the fair use inquiry, counsel for Oracle agreed that they are the "who, what, where, when, how, [and] how much." ***1194** Oral Arg. at 3:28–54, available at http://oralarguments.cafc.uscourts.gov/default.aspx?fl=2017-1118.mp3. Google did not dispute this characterization. This is, in part, because, in most fair use cases, defendants concede that they have used the copyrighted work, and "there is rarely dispute over the history, content, or origin of the copyrighted work." *See* Ned Snow, *Judges Playing Jury: Constitutional Conflicts in Deciding Fair Use on Summary Judgment*, 44 U.C. Davis L. Rev. 483, 493 (2010).

While some courts once treated the entire question of fair use as factual, and, thus, a question to be sent to the jury, that is not the modern view. [3] Since *Harper & Row*, the Ninth Circuit has described fair use as an "equitable defense." *Fisher v. Dees*, 794 F.2d 432, 435 (9th Cir. 1986) ("The fair-use doctrine was initially developed by courts as an equitable defense to copyright infringement."). Indeed, the Supreme Court referred to fair use as "an equitable rule of reason" in *Harper & Row*. 471 U.S. at 560, 105 S.Ct. 2218. Congress did the same when it codified the doctrine of fair use in 1976. *See* H.R. Rep. No. 94-1476, 94th Cong., 2d Sess. 65–66 (1976), U.S. Code Cong. & Admin. News 1976, 5659, 5679–80 ("[S]ince the doctrine [of fair use] is an equitable rule of reason, no generally applicable definition is possible, and each case raising the question must be decided on its own facts ...."). If fair use is equitable in nature, it would seem to be a question for the judge, not the jury, to decide, even when there are factual disputes regarding its application. *See Granite State Ins. Co. v. Smart Modular Techs., Inc.*, 76 F.3d 1023, 1027 (9th Cir. 1996) ("A litigant is not entitled to have a jury resolve

a disputed affirmative defense if the defense is equitable in nature."). In that instance, it would be the judge's factual determinations that would receive a deferential review—being assessed for clear error on the record before the court.

That said, the Supreme Court has never clarified whether and to what extent the jury is to play a role in the fair use analysis. *Harper & Row* involved an appeal from a *bench* trial where the district court concluded that the use of the copyrighted material was not a fair use. *Harper & Row Publishers, Inc. v. Nation Enters.*, 723 F.2d 195, 199 (2d Cir. 1983). The Court, thus, had no reason to discuss a jury determination of fair use and has not since taken an opportunity to do so.

Perhaps because of this silence, even after *Harper & Row*, several courts—including the Ninth Circuit—have continued to accept the fact that the question of fair use may go to a jury, albeit without analysis of why it may. *Compaq Comput. Corp. v. Ergonome Inc.*, 387 F.3d 403, 411 (5th Cir. 2004) ("The evidence presented at trial and the reasonable inferences therefrom, when viewed through the lens of the statutory fair use factors, support the jury's fair use finding."); *Jartech, Inc. v. Clancy*, 666 F.2d 403, 407–08 (9th Cir. 1982) (concluding that substantial evidence supported the jury's verdict on fair use); *Fiset v. Sayles*, No. 90-16548, 1992 WL 110263, at *4 (9th Cir. May 22, 1992) (finding that a reasonable jury could have concluded that "the evidence supporting fair use was not substantial"); *see also BUC Int'l Corp. v. Int'l Yacht Council*, 489 F.3d 1129, 1137 (11th Cir. 2007) (noting that the ***1195** fair use defense went to the jury); *N. Y. Univ. v. Planet Earth Found.*, 163 Fed.Appx. 13, 14 (2d Cir. 2005) ("As to the copyright infringement claim, the evidence also supports the jury's finding of fair use, under the four-factored analysis prescribed by statute.").

The Ninth Circuit has clarified, however, that the jury role in this context is limited to determining disputed "historical facts," not the inferences or conclusions to be drawn from those facts. *See Fisher*, 794 F.2d at 436. In *Fisher*, for example, the court explained that "[n]o material historical facts are at issue in this case. The parties dispute only the ultimate conclusions to be drawn from the admitted facts. Because, under *Harper & Row*, these judgments are legal in nature, we can make them without usurping the function of the jury." *Id.*; *see also Seltzer v. Green Day, Inc.*, 725 F.3d 1170, 1175 (9th Cir. 2013) ("As in *Fisher*, '[n]o material historical facts are at issue in this case. The parties dispute only the ultimate conclusion to be drawn from the admitted facts.' " (citing *Fisher*, 794 F.2d at 436)); *Hustler Magazine, Inc. v. Moral Majority, Inc.*, 606 F.Supp. 1526, 1532 (C.D. Cal. 1985) (noting that "fair use normally is a question of fact for the jury," but concluding that "the issue of fair use, at least in the context of this case, presents primarily a question of law"). Accordingly, while inferences from the four-factor analysis and the ultimate question of fair use are "legal in nature," in the Ninth Circuit, disputed historical facts represent questions for the jury. *Fisher*, 794 F.2d at 436. Where there are no disputed material historical facts, fair use can be decided by the court alone. *Id.*

Despite this case law, all aspects of Google's fair use defense went to the jury with neither party arguing that it should not. Thus, the jury was asked not just what the historical facts were, but what the implications of those facts were for the fair use defense. During the first appeal, Google argued to this court that there were disputed issues of material historical fact relevant to its fair use defense. As discussed below, the parties stipulated—or at least ceased to dispute—some of those facts, and presented the remaining disputed historical facts to the jury on remand. The jury returned a verdict in favor of Google on its fair use defense. Because the verdict form—though captioned as a "special verdict"—did not ask the jury to articulate its fact findings in any detail, we must assume that the jury resolved all factual issues relating to the historical facts in favor of the verdict.[4] Despite the posture of the fair use finding, we must break that finding into its constituent parts. We must then review the subsidiary and controverted findings of historical fact for substantial evidence. *See Seltzer*, 725 F.3d at 1175; *see also Brewer v. Hustler Magazine, Inc.*, 749 F.2d 527, 528 (9th Cir. 1984) ("We may disturb a jury verdict only if the evidence was insufficient as a matter of law.").

***1196** **[10]** All jury findings relating to fair use other than its implied findings of historical fact must, under governing Supreme Court and Ninth Circuit case law, be viewed as advisory only. Accordingly, while we might assess the jury's role in the assessment of fair use differently if not bound by Ninth Circuit law, we proceed on the assumption both that: (1) it was not error to send the question to the jury, because the Ninth Circuit has at least implicitly endorsed doing so; and (2) we must assess all inferences to be drawn from the historical facts found

by the jury and the ultimate question of fair use de novo, because the Ninth Circuit has explicitly said we must do so.

The parties have identified the following historical facts relating to Google's use of the copyrighted work:

- the history and origin of the copyrighted work, including what declaring code is;
- how much of the copyrighted work was copied;
- whether there were other ways to write the API packages;
- whether the copied material was used for the same purpose as in the original work;
- whether the use was commercial in nature;
- whether Google acted in bad faith in copying the work;
- whether there are functional aspects to the copyrighted work that make it less deserving of protection; and
- whether there was harm to the actual or potential markets for the copyrighted work.

The parties now agree on the resolution of the first four factual questions: (1) what the declaring code is and what it does in Java SE and Android, and that the code at issue was a work created by Oracle; (2) how many lines of code were copied; (3) that there were other ways for Google to write API packages; and (4) that Google used the API packages in Android for the same purpose they were created for in Java. The parties dispute, however, the remaining historical facts they identified. We address those disputes in the context of our assessment of the statutory factors to which the respective historical fact is relevant.

C. Applying the Fair Use Factors

Factor 1: The Purpose and Character of the Use

The first factor in the fair use inquiry involves "the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes." 17 U.S.C. § 107(1). This factor has two primary components: (1) whether the use is commercial in nature, rather than for educational or public interest purposes; and (2) "whether the new work is transformative or simply supplants the original." *Wall Data*, 447 F.3d at 778 (citing *Campbell*, 510 U.S. at 579, 114 S.Ct. 1164). As explained below, the first is a question of fact and the second is a question of law. As Oracle points out, moreover, courts sometimes also consider whether the historical facts support the conclusion that the infringer acted in bad faith. *See Harper & Row*, 471 U.S. at 562, 105 S.Ct. 2218. We address each component in turn.

a. Commercial Use

[11] Analysis of the first factor requires inquiry into the commercial nature of the use. Use of the copyrighted work that is commercial "tends to weigh against a finding of fair use." *Harper & Row*, 471 U.S. at 562, 105 S.Ct. 2218. Courts have recognized, however, that, "[s]ince many, if not most, secondary users seek at least ***1197** some measure of commercial gain from their use, unduly emphasizing the commercial motivation of a copier will lead to an overly restrictive view of fair use." *Am. Geophysical Union v. Texaco, Inc.*, 60 F.3d 913, 921 (2d Cir. 1994); *see also Infinity Broad. Corp. v. Kirkwood*, 150 F.3d 104, 109 (2d Cir. 1998) ("[N]otwithstanding its mention in the text of the statute, commerciality has only limited usefulness to a fair use inquiry; most secondary uses of copyrighted material, including nearly all of the uses listed in the statutory preamble, are commercial."). Accordingly, although the statute requires us to consider the "commercial nature" of the work, "the degree to which the new user exploits the copyright for commercial gain—as opposed to incidental use as part of a commercial enterprise—affects the weight we afford commercial nature as a factor." *Elvis Presley Enters., Inc. v. Passport Video*, 349 F.3d 622, 627 (9th Cir. 2003).

"[I]t is undisputed that Google's use of the declaring code and SSO from 37 Java API packages served commercial purposes." *Order Denying JMOL*, 2016 WL 3181206, at *7. Although the jury was instructed that commercial use weighed against fair use, the district court explained that the jury "could reasonably have found that Google's decision to make Android available open source and free for all to use had non-commercial purposes as well (such as the general interest in sharing software innovation)." *Id.*

On appeal, Oracle argues that Android is "hugely profitable" and that "Google reaps billions from exploiting Java in Android." Appellant Br. 29. As such, Oracle maintains that no reasonable jury could have found Android anything but "overwhelmingly commercial." *Id.* [5]

[12] Google responds that: (1) because it gives Android away for free under an open source license the jury could have concluded that Android has non-commercial purposes; and (2) the jury could have reasonably found that Google's revenue flows from the advertisements on its search engine which preexisted Android. Neither argument has merit.

First, the fact that Android is free of charge does not make Google's use of the Java API packages noncommercial. Giving customers "for free something they would ordinarily have to buy" can constitute commercial use. *A&M Records, Inc. v. Napster, Inc.*, 239 F.3d 1004, 1015 (9th Cir. 2001) (finding that "repeated and exploitative copying of copyrighted works, even if the copies are not offered for sale, may constitute a commercial use"). That Google might also have non-commercial motives is irrelevant as a matter of law. As the Supreme Court made clear when *The Nation* magazine published excerpts from Harper & Row's book, partly for the purpose of providing the public newsworthy information, the question "is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material ***1198** without paying the customary price." *Harper & Row*, 471 U.S. at 562, 105 S.Ct. 2218. Second, although Google maintains that its revenue flows from advertisements, not from Android, commerciality does not depend on how Google earns its money. Indeed, "[d]irect economic benefit is not required to demonstrate a commercial use." *A&M Records,* 239 F.3d at 1015. We find, therefore, that, to the extent we must assume the jury found Google's use of the API packages to be anything other than overwhelmingly commercial, that conclusion finds no substantial evidentiary support in the record. Accordingly, Google's commercial use of the API packages weighs against a finding of fair use.

b. Transformative Use

[13] Although the Copyright Act does not use the word "transformative," the Supreme Court has stated that the "central purpose" of the first fair use factor is to determine "whether and to what extent the new work is transformative." *Campbell*, 510 U.S. at 579, 114 S.Ct. 1164. Transformative works "lie at the heart of the fair use doctrine's guarantee of breathing space within the confines of copyright, and the more transformative the new work, the less will be the significance of other factors, like commercialism, that may weigh against a finding of fair use." *Id.* (internal citation omitted).

A use is "transformative" if it "adds something new, with a further purpose or different character, altering the first with new expression, meaning or message." *Id.* The critical question is "whether the new work merely supersede[s] the objects of the original creation ... or instead adds something new." *Id.* (citations and internal quotation marks omitted). This inquiry "may be guided by the examples given in the preamble to § 107, looking to whether the use is for criticism, or comment, or news reporting, and the like." *Id.* at 578–79, 114 S.Ct. 1164. "The Supreme Court has recognized that parodic works, like other works that comment and criticize, are by their nature often sufficiently transformative to fit clearly under the fair use exception." *Mattel, Inc. v. Walking Mountain Prods.*, 353 F.3d 792, 800 (9th Cir. 2003) (citing *Campbell*, 510 U.S. at 579, 114 S.Ct. 1164).

"Although transformation is a key factor in fair use, whether a work is transformative is a often highly contentious topic." *Seltzer*, 725 F.3d at 1176. Indeed, a "leading treatise on this topic has lamented the frequent misuse of the transformation test, complaining that it has become a conclusory label which is 'all things to all people.' " *Id.* (quoting Melville B. Nimmer & David Nimmer, 4 Nimmer on Copyright § 13.05[A][1][b], 13168–70 (2011)).

[14] **[15]** To be transformative, a secondary work must either alter the original with new expression, meaning, or message or serve a new purpose distinct from that of the original work. *Campbell,* 510 U.S. at 579, 114 S.Ct. 1164; *Elvis Presley Enters.*, 349 F.3d at 629. Where the use "is for the same intrinsic purpose as [the copyright holder's] ... such use seriously weakens a claimed fair use." *Worldwide Church of God v. Phila. Church of God, Inc.*, 227 F.3d 1110, 1117 (9th Cir. 2000) (quoting *Weissmann v. Freeman*, 868 F.2d 1313, 1324 (2d Cir. 1989)).

[16] Although "transformative use is not absolutely necessary for a finding of fair use, the goal of copyright, to promote science and the arts, is generally furthered by the creation of transformative works." *Campbell*, 510 U.S. at 579, 114 S.Ct. 1164 (citation and footnote omitted). As such, "the more transformative the new work, the less will be the significance of other factors, like commercialism, that may weigh against a finding of fair use." *Id.* ***1199** Importantly, in the Ninth Circuit, whether a work is transformative is a question of law. *See Mattel,* 353 F.3d at 801 (explaining that parody—a well-established species of transformative use—"is a question of law, not a matter of public majority opinion"); *see also Fox News Network, LLC v. TVEyes, Inc.*, 883 F.3d 169, 175–77 (2d Cir. 2018) (reassessing whether the use in question was transformative and deciding it was as a matter of law).

In denying JMOL, the district court explained that "of course, the copied declarations serve the same function in both works, for by definition, declaring code in the Java programming language serves the [same] specific definitional purposes." *Order Denying JMOL*, 2016 WL 3181206, at *8.[6] The court concluded, however, that the jury could reasonably have found that Google's selection of some, but not all, of the Java API packages—"with new implementing code adapted to the constrained operating environment of mobile smartphone devices," together with new "methods, classes, and packages written by Google for the mobile smartphone platform"—constituted "a fresh context giving new expression, meaning, or message to the duplicated code." *Id.* at *9.

On appeal, Oracle argues that Google's use was not transformative because it did not alter the APIs with "new expression, meaning, or message." Appellant Br. 29 (quoting *Campbell*, 510 U.S. at 579, 114 S.Ct. 1164). Because Google concedes that it uses the API packages for the same purpose, Oracle maintains that it was unreasonable for either the jury or the court to find that Google sufficiently transformed the APIs to overcome its highly commercial use.

Google responds that a reasonable jury could have concluded that Google used a small portion of the Java API packages to create a new work in a new context —"Android, a platform for smartphones, not desktops and servers." Cross–Appellant Br. 37. Google argues that, although the declarations and SSO may perform the same functions in Android and Java, the jury could reasonably find that they have different purposes because the "point of Android was to create a groundbreaking platform for smartphones." *Id.* at 39.

[17] Google's arguments are without merit. As explained below, Google's use of the API packages is not transformative as a matter of law because: (1) it does not fit within the uses listed in the preamble to § 107; (2) the purpose of the API packages in Android is the same as the purpose of the packages in the Java platform; (3) Google made no alteration to the expressive content or message of the copyrighted material; and (4) smartphones were not a new context.

First, though not dispositive, we turn to the examples given in the preamble to § 107, "looking to whether the use is for ***1200** criticism, or comment, or news reporting, and the like." *Campbell*, 510 U.S. at 578–79, 114 S.Ct. 1164. Google's use of the Java API packages does not fit within the statutory categories, and Google does not suggest otherwise. Instead, Google cites *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000), for the proposition that the "Ninth Circuit has held other types of uses—specifically including uses of computer code—to be fair." Cross–Appellant Br. 41. In *Sony*, the court found that the defendant's reverse engineering and intermediate copying of Sony's copyrighted software system "was a fair use for the purpose of gaining access to the unprotected elements of Sony's software." 203 F.3d at 602. The court explained that Sony's software program contained unprotected functional elements and that the defendant could only access those elements through reverse engineering. *Id.* at 603. The defendant used that information to create a software program that let consumers play games designed for Sony's PlayStation console on their computers. The court found that the defendant's use was only "modestly transformative" where: (1) the defendant created "a wholly new product" with "entirely new ... code," and (2) the intermediate copying was performed to "produce a product that would be compatible." *Id.* at 606–07. As Oracle points out, even the "modest" level of transformation at issue in *Sony* is more transformative than what Google did here: copy code verbatim to attract programmers to Google's "new and incompatible platform." Appellant Response Br. 21.

It is undisputed that the API packages "serve the same function in both works." *Order Denying JMOL*, 2016 WL 3181206, at *8. And, as Oracle explains, the historical facts relevant to transformative use are also undisputed: what declaring code is, what it does in Java and in Android, how the audience of computer developers perceives it, how much Google took and added, what the added code does, and why Google used the declaring code and SSO. Indeed, Google conceded that "including the declarations (and their associated SSO) was for the benefit of developers, who—familiar with the Java programming language—had certain expectations regarding the language's APIs." Google's Opp. to Oracle's Rule 50(a) Motion for JMOL at 20, *Oracle Am., Inc. v. Google Inc.*, No. 3:10-cv-3561 (N.D. Cal. May 21, 2016), ECF No. 1935. The fact that Google created exact copies of the declaring code and SSO and used those copies for the same purpose as the original material "seriously weakens [the] claimed fair use." *See Wall Data*, 447 F.3d at 778 (finding that, where the "Sheriff's Department created exact copies of RUMBA's software ... [and] put those copies to the identical purpose as the original software," the use was not transformative); *see also Campbell*, 510 U.S. at 580, 114 S.Ct. 1164 (noting that where the alleged infringer merely seeks "to avoid the drudgery in working up something fresh," any "claim to fairness ... diminishes accordingly").

Google argues that Android is transformative because Google selectively used the declarations and SSO of only 37 of the 166 Java SE API packages and wrote its own implementing code. But taking only select passages of a copyrighted work is, by itself, not transformative. *See L.A. News Serv. v. CBS Broad., Inc.*, 305 F.3d 924, 938–39 (9th Cir. 2002) ("Merely plucking the most visually arresting excerpt from LANS's nine minutes of footage cannot be said to have added anything new."). While, as discussed below, the volume of work copied is relevant to the fair use inquiry generally, thought must be given to the quality and importance of the copied material, not just to its relative quantity vis-à-vis the overall work. *See* ***1201** *Campbell*, 510 U.S. at 586–87, 114 S.Ct. 1164. To hold otherwise would mean that verbatim copying could qualify as fair use as long as the plagiarist stops short of taking the entire work. That approach is inconsistent with settled law and is particularly troubling where, as here, the portion copied is qualitatively significant. *See Harper & Row*, 471 U.S. at 569, 105 S.Ct. 2218 (finding that verbatim copying of 300 words from a manuscript of more than 200,000 words was not a fair use); *see also Folsom v. Marsh*, 9 F. Cas. 342, 345 (C.C.D. Mass 1841) (Story, J.) ("There must be real, substantial condensation of the materials, and intellectual labor and judgment bestowed thereon; and not merely the facile use of the scissors; or extracts of the essential parts, constituting the chief value of the original work.").

That Google wrote its own implementing code is irrelevant to the question of whether use of the APIs was transformative. As we noted in the prior appeal, "no plagiarist can excuse the wrong by showing how much of his work he did not pirate." *Oracle*, 750 F.3d at 1375 (quoting *Harper & Row*, 471 U.S. at 565, 105 S.Ct. 2218). The relevant question is whether Google altered "the *expressive content or message* of the original work" that it copied—not whether it rewrote the portions it did not copy. *See Seltzer*, 725 F.3d at 1177 (explaining that a work is not transformative where the user "makes no alteration to the *expressive content or message* of the original work"). That said, even where the allegedly infringing work "makes few physical changes to the original or fails to comment on the original," it will "typically [be] viewed as transformative as long as new expressive content or message is apparent." *Id.* Here, however, there is no suggestion that the new implementing code somehow changed the expression or message of the declaring code. While Google's use could have been transformative if it had copied the APIs for some other purpose—such as teaching how to design an API—merely copying the material and moving it from one platform to another without alteration is not transformative.

Google's primary argument on appeal is that Android is transformative because Google incorporated the declarations and SSO of the 37 API packages into a new *context*—smartphones. But the record showed that Java SE APIs were in smartphones before Android entered the market. Specifically, Oracle presented evidence that Java SE was in SavaJe mobile phones and that Oracle licensed Java SE to other smartphone manufacturers, including Danger and Nokia. Because the Java SE was already being used in smartphones, Google did not "transform" the copyrighted material into a new context and no reasonable jury could conclude otherwise. [7]

[18] In any event, moving material to a new context is not transformative in and of itself—even if it is a "sharply different context." *TCA Television Corp. v. McCollum*, 839 F.3d 168, 181–83 (2d Cir. 2016) (finding that use

"at some length, almost verbatim," of the copyrighted comedy routine "Who's on First?" in a dramatic play was not transformative where the play neither "imbued the Routine with any new expression, meaning, or message," nor added "any new dramatic purpose"). As previously explained, a use becomes transformative only if it serves a different purpose or alters the "expression, meaning, or message" of the original work. ***1202** *Kelly*, 336 F.3d at 818. As such, "[c]ourts have been reluctant to find fair use when an original work is merely retransmitted in a different medium." *A&M Records,* 239 F.3d at 1015. Accordingly, although a change of format may be "useful," it "is not technically a transformation." *Infinity Broad.*, 150 F.3d at 108 n.2 (finding that retransmitting copyrighted radio transmissions over telephone lines was not transformative because there was no new expression, meaning, or message).

The Ninth Circuit has stated that "[a] use is considered transformative only where a defendant changes a plaintiff's copyrighted work or uses the plaintiff's copyrighted work in a different context such that the plaintiff's work is transformed into a new creation." *Perfect 10, Inc. v. Amazon.com, Inc.*, 508 F.3d 1146, 1165 (9th Cir. 2007) (quoting *Wall Data*, 447 F.3d at 778). In *Perfect 10*, for example, the court found Google's use of thumbnail versions of copyrighted images "highly transformative" because, "[a]lthough an image may have been created originally to serve an entertainment, aesthetic, or informative function, a search engine transforms the image into a pointer directing a user to a source of information." *Id.* Although the court discussed the change in context (moving the copyrighted images into the electronic reference tool), it emphasized that Google used the images "in a new context to serve a different purpose." *Id.* In reaching this conclusion, the court reiterated that "even making an exact copy of a work may be transformative so long as the copy serves a different function than the original work." *Id.* (citing *Kelly*, 336 F.3d at 818–19). It is clear, therefore, that the change in context alone was not dispositive in *Perfect 10*; rather, the change in context facilitated the change in purpose, which made the use transformative.

To some extent, any use of copyrighted work takes place in a slightly different context than the original. And of course, there is no bright line identifying when a use becomes transformative. But where, as here, the copying is verbatim, for an identical function and purpose, and there are no changes to the expressive content or message, a mere change in format (e.g., from desktop and laptop computers to smartphones and tablets) is insufficient as a matter of law to qualify as a transformative use. [8]

c. Bad faith

In evaluating the "purpose and character" factor, the Ninth Circuit applies "the general rule that a party claiming fair use must act in a manner generally compatible with principles of good faith and fair dealing." *Perfect 10,* 508 F.3d at 1164 n.8 (citing *Harper & Row*, 471 U.S. at 562–63, 105 S.Ct. 2218). In part, this is based on the fact that, in *Harper & Row,* the Supreme Court expressly stated that "[f]air use presupposes 'good faith' and 'fair dealing.' " 471 U.S. at 562, 105 S.Ct. 2218 (citation omitted). It is also in part true because, as the Ninth Circuit has said, one who acts in bad faith should be barred from invoking the equitable defense of fair use. *Fisher*, 794 F.2d at 436 (calling the principle of considering the alleged infringer's "bad conduct" as a "bar [to] his use of the equitable defense of fair use" a sound one). [9]

***1203** Consistent with this authority, and at Oracle's request, the district court instructed the jury that it could consider whether Google acted in bad faith (or not) as part of its assessment of the first fair use factor. *Order Denying JMOL*, 2016 WL 3181206, at *6. And, because Oracle was permitted to introduce evidence that Google acted in bad faith, the court permitted Google to try to prove its good faith. *Id.*

At trial, Oracle introduced evidence suggesting that "Google felt it needed to copy the Java API as an accelerant to bring Android to the market quicker" and knew that it needed a license to use Java. *Id.* For its part, Google presented evidence that it believed that the declaring code and SSO were "free to use and reimplement, both as a matter of developer practice and because the availability of independent implementations of the Java API enhanced the popularity of the Java programming language, which Sun promoted as free for all to use." *Id.* at *7. Given this conflicting evidence, the district court found that the jury could reasonably have concluded that "Google's use of parts of the Java API as an accelerant was undertaken based on a good faith belief that at least the declaring code and SSO were free to use

(which it did use), while a license was necessary for the implementing code (which it did not use)." *Id.*

On appeal, Oracle argues that there was ample evidence that Google intentionally copied Oracle's copyrighted work and knew that it needed a license to use Java. Google responds that the jury heard sufficient evidence of Google's good faith based on industry custom and was entitled to credit that evidence.

[19] **[20]** But, while bad faith may weigh against fair use, a copyist's good faith cannot weigh in favor of fair use. Indeed, the Ninth Circuit has expressly recognized that "the innocent intent of the defendant constitutes no defense to liability." *Monge*, 688 F.3d at 1170 (quoting 4 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 13.08[B][1] (Matthew Bender rev. ed. 2011)). If it were clear, accordingly, that the jury found fair use solely or even largely because it *approved* of Google's motives even if they were in bad faith, we would find such a conclusion improper. Because evidence of Google's good faith was relevant to rebut evidence of its bad faith, however, and there is no objection to the instructions to the jury on this or any other point, we must assume that the jury simply did not find the evidence of Google's bad faith persuasive.[10] We note, ***1204** moreover, that merely "being denied permission to use a work does not weigh against a finding of fair use." *Campbell*, 510 U.S. at 585 n.18, 114 S.Ct. 1164 ("If the use is otherwise fair, then no permission need be sought or granted.").

Ultimately, we find that, even assuming the jury was unpersuaded that Google acted in bad faith, the highly commercial and non-transformative nature of the use strongly support the conclusion that the first factor weighs against a finding of fair use.

Factor 2: Nature of the Copyrighted Work

[21] **[22]** The second factor—the nature of the copyrighted work—"calls for recognition that some works are closer to the core of intended copyright protection than others, with the consequence that fair use is more difficult to establish when the former works are copied." *Campbell*, 510 U.S. at 586, 114 S.Ct. 1164. This factor "turns on whether the work is informational or creative." *Worldwide Church of God*, 227 F.3d at 1118; *see also Harper & Row*, 471 U.S. at 563, 105 S.Ct. 2218 ("The law generally recognizes a greater need to disseminate factual works than works of fiction or fantasy."). Creative expression "falls within the core of the copyright's protective purposes." *Campbell*, 510 U.S. at 586, 114 S.Ct. 1164. Although "software products are not purely creative works," it is well established that copyright law protects computer software. *Wall Data*, 447 F.3d at 780 (citing *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1519 (9th Cir. 1992) ("[T]he 1980 amendments to the Copyright Act unambiguously extended copyright protection to computer programs.")).

Here, the district court found that the jury could have concluded that the process of designing APIs was "highly creative" and "thus at the core of copyright's protection" or it could "reasonably have gone the other way and concluded that the declaring code was not highly creative." *Order Denying JMOL*, 2016 WL 3181206, at *10. While the jury heard testimony from Google's own expert that API design is "an art, not a science," other witnesses emphasized the functional role of the declaring code and the SSO and minimized the creative aspects. *Id.* Accordingly, the district court concluded that the "jury could reasonably have found that, while the declaring code and SSO were creative enough to qualify for copyright protection, functional considerations predominated in their design." *Id.*

On appeal, Oracle emphasizes that designing the APIs was a highly creative process and that the organization of the packages was not mandated by function. Indeed, this court has already held that the declaring code and the SSO of the 37 API packages at issue were sufficiently creative and original to qualify for copyright protection. *Oracle*, 750 F.3d at 1356. According to Oracle, the district court erred in assuming that, because the APIs have a "functional role," they cannot be creative.

As Google points out, however, all we found in the first appeal was that the declarations and SSO were sufficiently creative to provide the "minimal degree of creativity," *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991), that is required for copyrightability. We also recognized that a reasonable jury could find that "the functional aspects of the packages" ***1205** are "relevant to Google's fair use defense." *Oracle*, 750 F.3d at 1369, 1376–77. On remand, Oracle stipulated that some of the declarations were necessary to use the Java language and presented

no evidence explaining how the jury could distinguish the functionality and creativity of those declarations from the others. Google maintains that it presented evidence that the declarations and SSO were functional and the jury was entitled to credit that evidence.

Although it is clear that the 37 API packages at issue involved some level of creativity—and no reasonable juror could disagree with that conclusion—reasonable jurors could have concluded that functional considerations were both substantial and important. Based on that assumed factual finding, we conclude that factor two favors a finding of fair use.

The Ninth Circuit has recognized, however, that this second factor "typically has not been terribly significant in the overall fair use balancing." *Dr. Seuss Enters., L.P. v. Penguin Books USA, Inc.*, 109 F.3d 1394, 1402 (9th Cir. 1997) (finding that the "creativity, imagination and originality embodied in The Cat in the Hat and its central character tilts the scale against fair use"); *Mattel*, 353 F.3d at 803 (similar). Other circuits agree. *Fox News Network*, 883 F.3d at 178 ("This factor 'has rarely played a significant role in the determination of a fair use dispute,' and it plays no significant role here." (quoting *Authors Guild v. Google, Inc.*, 804 F.3d 202, 220 (2d Cir. 2015))). We note, moreover, that allowing this one factor to dictate a conclusion of fair use in all cases involving copying of software could effectively negate Congress's express declaration—continuing unchanged for some forty years—that software is copyrightable. Accordingly, though the jury's assumed view of the nature of the copyrighted work weighs in favor of finding fair use, it has less significance to the overall analysis.

Factor 3: Amount and Substantiality of the Portion Used

[23] **[24]** **[25]** The third factor focuses on the "amount and substantiality of the portion used in ... the context of the copyrighted work, not the infringing work." *Oracle*, 750 F.3d at 1375. Indeed, the statutory language makes clear that "a taking may not be excused merely because it is insubstantial with respect to the *infringing* work." *Harper & Row*, 471 U.S. at 565, 105 S.Ct. 2218. "[T]he fact that a substantial portion of the infringing work was copied verbatim [from the original work] is evidence of the qualitative value of the copied material, both to the originator and to the plagiarist who seeks to profit from marketing someone else's copyrighted expression." *Id.* Thus, while "whole-sale copying does not preclude fair use per se, copying an entire work militates against a finding of fair use." *Worldwide Church of God*, 227 F.3d at 1118 (citation and quotation marks omitted). But, there is no relevance to the opposite—i.e., adding substantial content to the copyrighted work is not evidence that what was copied was insubstantial or unimportant.

The inquiry under this third factor "is a flexible one, rather than a simple determination of the percentage of the copyrighted work used." *Monge*, 688 F.3d at 1179. The Ninth Circuit has explained that this third factor looks to the quantitative amount and qualitative value of the original work used in relation to the justification for its use. *Seltzer*, 725 F.3d at 1178. The percentage of work copied is not dispositive where the portion copied was qualitatively significant. *Harper & Row*, 471 U.S. at 566, 105 S.Ct. 2218 ("In view of the expressive value of the excerpts and their key role in the infringing work, we cannot ***1206** agree with the Second Circuit that the 'magazine took a meager, indeed an infinitesimal amount of Ford's original language.' " (citation omitted)). Google is correct that the Ninth Circuit has said that, "this factor will not weigh against an alleged infringer, even when he copies the whole work, if he takes no more than is necessary for his intended use." *Id.* (citing *Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 820–21 (9th Cir. 2003)). But the Ninth Circuit has only said that is true where the intended use was a transformative one, because the "extent of permissible copying varies with the purpose and character of the use." *Id.* (quoting *Campbell*, 510 U.S. at 586–87, 114 S.Ct. 1164). Here, we have found that Google's use was not transformative and Google has conceded both that it could have written its own APIs and that the purpose of its copying was to make Android attractive to programmers. "Necessary" in the context of the cases upon which Google relies does not simply mean easier.

In assessing factor three, the district court explained that the "jury could reasonably have found that Google duplicated the bare minimum of the 37 API packages, just enough to preserve inter-system consistency in usage, namely the declarations and their SSO only, and did not copy any of the implementing code," such that Google "copied only so much as was reasonably necessary." *Order Denying JMOL*, 2016 WL 3181206, at *10. In reaching this conclusion, the court noted that the jury could have

found that the number of lines of code Google duplicated was a "tiny fraction of one percent of the copyrighted works (and even less of Android, for that matter)." *Id.* We disagree that such a conclusion would have been reasonable or sufficient on this record.

On remand, the parties stipulated that only 170 lines of code were necessary to write in the Java language. It is undisputed, however, that Google copied 11,500 lines of code—11,330 more lines than necessary to write in Java. That Google copied more than necessary weighs against fair use. *See Monge*, 688 F.3d at 1179 (finding that, where the copyist "used far more than was necessary" of the original work, "this factor weighs against fair use"). And, although Google emphasizes that it used a small percentage of Java (11,500 lines of declarations out of roughly 2.86 million lines of code in the Java SE libraries), it copied the SSO for the 37 API packages in its entirety.

[26] **[27]** The district court emphasized Google's desire to "preserve inter-system consistency" to "avoid confusion among Java programmers as between the Java system and the Android system." *Order Denying JMOL*, 2016 WL 3181206, at *10–11. As we noted in the prior appeal, however, Google did not seek to foster any "inter-system consistency" between its platform and Oracle's Java platform. *Oracle*, 750 F.3d at 1371. And Google does not rely on any interoperability arguments in this appeal.[11] Google sought "to capitalize on the fact that software developers were already trained and experienced in using the Java API packages at issue." *Id.* But there is no inherent right to copy in order to capitalize on the popularity of the copyrighted ***1207** work or to meet the expectations of intended customers. Taking those aspects of the copyrighted material that were familiar to software developers to create a similar work designed to be popular with those same developers is not fair use. *See Dr. Seuss Enters.*, 109 F.3d at 1401 (copying the most famous and well recognized aspects of a work "to get attention" or "to avoid the drudgery in working up something fresh" is not a fair use (quoting *Campbell*, 510 U.S. at 580, 114 S.Ct. 1164)).

Even assuming the jury accepted Google's argument that it copied only a small portion of Java, no reasonable jury could conclude that what was copied was qualitatively insignificant, particularly when the material copied was important to the creation of the Android platform. Google conceded as much when it explained to the jury the importance of the APIs to the developers it wished to attract. *See* Tr. of Proceedings held on 5/16/16 at 106:8-14, *Oracle Am., Inc. Google Inc.*, No. 3:10-cv-3561 (N.D. Cal. May 20, 2016), ECF No. 1930; *Id.* at 134:6–11. Indeed, Google's own expert conceded that "it was a sound business practice for Google to leverage the existing community of developers, minimizing the amount of new material and maximizing existing knowledge," even though Google also conceded that it could have written the APIs differently to achieve the same functions. *Id.* at 144:5–10. For these reasons, we find that the third factor is, at best, neutral in the fair use inquiry, and arguably weighs against such a finding.

Factor 4: Effect Upon the Potential Market

[28] The fourth and final factor focuses on "the effect of the use upon the potential market for or value of the copyrighted work." 17 U.S.C. § 107(4). This factor reflects the idea that fair use "is limited to copying by others which does not materially impair the marketability of the work which is copied." *Harper & Row*, 471 U.S. at 566–67, 105 S.Ct. 2218. It requires that courts "consider not only the extent of market harm caused by the particular actions of the alleged infringer, but also whether unrestricted and widespread conduct of the sort engaged in by the defendant ... would result in a substantially adverse impact on the potential market for the original." *Campbell*, 510 U.S. at 590, 114 S.Ct. 1164 (citation and quotation marks omitted).

The Supreme Court once said that factor four is "undoubtedly the single most important element of fair use." *Harper & Row*, 471 U.S. at 566, 105 S.Ct. 2218. In its subsequent opinion in *Campbell*, however, the Court emphasized that none of the four factors can be viewed in isolation and that "[a]ll are to be explored, and the results weighed together, in light of the purposes of copyright." 510 U.S. at 578, 114 S.Ct. 1164; *see also Infinity Broad.*, 150 F.3d at 110 ("Historically, the fourth factor has been seen as central to fair use analysis, although the Supreme Court appears to have backed away from this position." (internal citation omitted)). The Court has also explained that "[m]arket harm is a matter of degree, and the importance of this factor will vary, not only with the amount of harm, but also with the relative strength of the showing on the other factors." *Campbell*, 510 U.S. at 590 n.21, 114 S.Ct. 1164.

The Ninth Circuit recently indicated that likely market harm can be presumed where a use is "commercial and not transformative." *Disney Enters., Inc. v. VidAngel, Inc.*, 869 F.3d 848, 861 (9th Cir. 2017) (citing *Leadsinger*, 512 F.3d at 531, for the proposition that, where a use "was commercial and not transformative, it was not error to presume likely market harm"). ***1208** That presumption allegedly traces back to *Sony Corp. of America v. University City Studios, Inc.*, 464 U.S. 417, 451, 104 S.Ct. 774, 78 L.Ed.2d 574 (1984), where the Supreme Court stated that, "[i]f the intended use is for commercial gain, that likelihood [of future harm] may be presumed. But if it is for a noncommercial purpose, the likelihood must be demonstrated." The Supreme Court has since clarified that market impact, "no less than the other three [factors], may be addressed only through a 'sensitive balancing of interests' " and that earlier interpretations of *Sony* to the contrary were incorrect. *Campbell*, 510 U.S. at 590 n.21, 114 S.Ct. 1164 (quoting *Sony*, 464 U.S. at 455 n.40, 104 S.Ct. 774);[12] *see also Monge*, 688 F.3d at 1181 (cautioning against overemphasis on a presumption of market harm after *Campbell*). On this point, we must apply clear Supreme Court precedent rather than the more recent Ninth Circuit's statements to the contrary.

[29] **[30]** In evaluating the fourth factor, courts consider not only harm to the actual or potential market for the copyrighted work, but also harm to the "market for potential derivative uses," including "those that creators of original works would in general develop or license others to develop." *Campbell*, 510 U.S. at 592, 114 S.Ct. 1164; *see also A&M Records*, 239 F.3d at 1017 ("[L]ack of harm to an established market cannot deprive the copyright holder of the right to develop alternative markets for the works."). A court can therefore consider the challenged use's "impact on potential licensing revenues for traditional, reasonable, or likely to be developed markets." *Swatch Grp. Mgmt. Servs. Ltd. v. Bloomberg L.P.*, 756 F.3d 73, 91 (2d Cir. 2014) (citation omitted); *see also Seltzer*, 725 F.3d at 1179 ("This factor also considers any impact on 'traditional, reasonable, or likely to be developed markets.' " (citation omitted)).

[31] Also relevant to the inquiry is the fact that a copyright holder has the exclusive right to determine "when, 'whether and in what form to release' " the copyrighted work into new markets, whether on its own or via a licensing agreement. *Monge*, 688 F.3d at 1182 (quoting *Harper & Row*, 471 U.S. at 553, 105 S.Ct. 2218). Indeed, the Ninth Circuit has recognized that "[e]ven an author who had disavowed any intention to publish his work during his lifetime" was entitled to copyright protection because: (1) "the relevant consideration was the 'potential market' " and (2) "he has the right to change his mind." *Worldwide Church*, 227 F.3d at 1119 (citing *Salinger v. Random House, Inc.*, 811 F.2d 90, 99 (2d Cir. 1987)); *see also Micro Star v. Formgen Inc.*, 154 F.3d 1107, 1113 (9th Cir. 1998) (noting that only the copyright holder "has the right to enter that market; whether it chooses to do so is entirely its business").

Here, the district court concluded 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." *Order Denying JMOL*, 2016 WL 3181206, at *10. In reaching this conclusion, the district court noted that, before Android was released, Sun made all of the Java API packages available for free and open source under the name OpenJDK, subject only to the terms of a general ***1209** public license. *Id.* According to the district court, the jury could have concluded that "Android's impact on the market for the copyrighted works paralleled what Sun already expected via its OpenJDK." *Id.*

[32] On appeal, Oracle argues that the evidence of actual and potential harm stemming from Google's copying was "overwhelming," and that the district court erred as a matter of law in concluding otherwise. Appellant Br. 52. We agree.

First, with respect to actual market harm, the evidence showed that Java SE had been used for years in mobile devices, including early smartphones, prior to Android's release. Specifically, the jury heard testimony that Java SE was already in smartphones, including Blackberry, SavaJe, Danger, and Nokia. That Android competed directly with Java SE in the market for mobile devices is sufficient to undercut Google's market harm arguments. With respect to tablets, the evidence showed that Oracle licensed Java SE for the Amazon Kindle. After Android's release, however, Amazon was faced with two competing options—Java SE and Android—and selected Android.[13] The jury also heard evidence that Amazon later used the fact that Android was free to negotiate a

steep discount to use Java SE in its newer e-reader. In other words, the record contained substantial evidence that Android was used as a substitute for Java SE and had a direct market impact. Given this evidence of actual market harm, no reasonable jury could have concluded that there was no market harm to Oracle from Google's copying.

Even if there were a dispute about whether Oracle was licensing Java SE in smartphones at the time Android launched, moreover, "fair use focuses on *potential*, not just actual, market harm." *Monge*, 688 F.3d at 1181. Accordingly, although the district court focused exclusively on the market it found that Oracle had already entered—desktops and laptops—it should have considered how Google's copying affected potential markets Oracle might enter or derivative works it might create or license others to create. *See Campbell*, 510 U.S. at 590, 114 S.Ct. 1164. Licensing Java SE for smartphones with increased processing capabilities was one such potential new market. And the fact that Oracle and Google engaged in lengthy licensing negotiations demonstrates that Oracle was attempting to license its work for mobile devices, including smartphones. [14] Smartphones were, therefore, a "traditional, reasonable, or likely to be developed market." *See Swatch Grp.*, 756 F.3d at 91; *see also Seltzer*, 725 F.3d at 1179.

Google argues that a reasonable jury could have concluded that Java SE and Android did not compete in the same market because Oracle: (1) was not a device maker; and (2) had not yet built its own smartphone platform. Neither argument has merit. That Oracle never built a smartphone ***1210** device is irrelevant because potential markets include licensing others to develop derivative works. *See Campbell*, 510 U.S. at 592, 114 S.Ct. 1164. The fact that Oracle had not yet developed a smartphone platform is likewise irrelevant as a matter of law because, as Oracle submits, a market is a potential market even where the copyright owner has no immediate plans to enter it or is unsuccessful in doing so. *See Worldwide Church*, 227 F.3d at 1119; *Micro Star*, 154 F.3d at 1113. Even assuming a reasonable jury could have found no *current* market harm, the undisputed evidence showed, at a minimum, that Oracle intended to license Java SE in smartphones; there was no evidence in the record to support any contrary conclusion. Because the law recognizes and protects a copyright owner's right to enter a "potential market," this fact alone is sufficient to establish market impact.

Given the record evidence of actual and potential harm, we conclude that "unrestricted and widespread conduct of the sort engaged in by" Google would result in "a substantially adverse impact on the potential market for the original" and its derivatives. *See Campbell*, 510 U.S. at 590, 114 S.Ct. 1164 (citation and quotation marks omitted). Accordingly, the fourth factor weighs heavily in favor of Oracle.

Balancing the Four Factors

[33] Having undertaken a case-specific analysis of all four factors, we must weigh the factors together "in light of the purposes of copyright." *Campbell*, 510 U.S. at 578, 114 S.Ct. 1164. We conclude that allowing Google to commercially exploit Oracle's work will not advance the purposes of copyright in this case. Although Google could have furthered copyright's goals of promoting creative expression and innovation by developing its own APIs, or by licensing Oracle's APIs for use in developing a new platform, it chose to copy Oracle's creative efforts instead. There is nothing fair about taking a copyrighted work verbatim and using it for the same purpose and function as the original in a competing platform.

Even if we ignore the record evidence and assume that Oracle was not already licensing Java SE in the smartphone context, smartphones were undoubtedly a potential market. Android's release effectively replaced Java SE as the supplier of Oracle's copyrighted works and prevented Oracle from participating in developing markets. This superseding use is inherently unfair.

On this record, factors one and four weigh heavily against a finding of fair use, while factor two weighs in favor of such a finding and factor three is, at best, neutral. Weighing these factors together, we conclude that Google's use of the declaring code and SSO of the 37 API packages was not fair as a matter of law.

We do not conclude that a fair use defense could never be sustained in an action involving the copying of computer code. Indeed, the Ninth Circuit has made it clear that some such uses can be fair. *See Sony*, 203 F.3d at 608; *Sega*, 977 F.2d at 1527–28. We hold that, given the facts relating

to the copying at issue here—which differ materially from those at issue in *Sony* and *Sega*—Google's copying and use of this particular code was not fair as a matter of law.

III. GOOGLE'S CROSS–APPEAL

Google cross-appeals from the district court's final judgment solely to "preserv[e] its claim that the declarations/SSO are not protected by copyright law." Cross–Appellant Br. 83. Specifically, Google maintains that the declaring code and SSO are: (1) an unprotected "method of operation" under 17 U.S.C. § 102(b), because they allow ***1211** programmers to operate the pre-written programs of the Java language; and (2) subject to the merger doctrine. We resolved these issues against Google in the first appeal, finding that the declaring code and the SSO of the 37 API packages at issue are entitled to copyright protection. *Oracle*, 750 F.3d at 1354.

Google did not petition this court for rehearing and instead filed a petition for a writ of certiorari asking the Supreme Court to determine whether our copyrightability determination was in error. Oracle responded to the petition, and the Supreme Court invited the Solicitor General to express the views of the United States. The government agreed that Oracle's computer code is copyrightable, and the Supreme Court denied Google's petition in June 2015. *Google, Inc. v. Oracle Am., Inc.*, —— U.S. ——, 135 S.Ct. 2887, 192 L.Ed.2d 948 (2015).

Google neither asks the panel for relief on the copyrightability issue nor offers any arguments on that issue. We remain convinced that our earlier copyrightability decision was consistent with Congress's repeated directives on the subject. Accordingly, we provide no relief to Google on its cross-appeal, finding a ruling on it unnecessary.

IV. CONCLUSION

For the foregoing reasons, we conclude that Google's use of the 37 Java API packages was not fair as a matter of law. We therefore reverse the district court's decisions denying Oracle's motions for JMOL and remand for a trial on damages. The district court may determine the appropriate vehicle for consideration of infringement allegations regarding additional uses of Android. We dismiss Google's cross-appeal.

REVERSED AND REMANDED; CROSS–APPEAL DISMISSED

COSTS

No costs.

All Citations

886 F.3d 1179, 2018 Copr.L.Dec. P 31,248, 126 U.S.P.Q.2d 1228

Footnotes

1 In September 2017, Google converted from a corporation to a limited liability company and changed its name to Google LLC, as reflected in the amended caption.

2 The jury found no patent infringement, and the patent claims are not at issue on appeal.

3 In *DC Comics, Inc. v. Reel Fantasy, Inc.*, 696 F.2d 24, 28 (2d Cir. 1982), the Second Circuit found that "[t]he four factors listed in Section 107 raise essentially factual issues and, as the district court correctly noted, are normally questions for the jury." So too, Justice Joseph Story described fair use as a "question of fact to come to a jury" in 1845. *Emerson v. Davies*, 8 F. Cas. 615, 623–24 (C.C.D. Mass. 1845).

4 As counsel for Oracle noted at oral argument, this is similar to the standard we apply in obviousness cases. Oral Argument at 9:34–10:24. Because obviousness is a mixed question of law and fact, we "first presume that the jury resolved the underlying factual disputes in favor of the verdict [ ] and leave those presumed findings undisturbed if they are supported by substantial evidence. Then we examine the [ultimate] legal conclusion [of obviousness] de novo to see whether it is correct in light of the presumed jury fact findings." *Kinetic Concepts, Inc. v. Smith & Nephew, Inc.*, 688 F.3d 1342, 1356–57 (Fed. Cir. 2012) (quoting *Jurgens v. McKasy*, 927 F.2d 1552, 1557 (Fed. Cir. 1991)). Likewise, Google cited our decision in *Kinetic Concepts* for the proposition that we must "presume that the jury made all findings in support of the verdict that are supported by substantial evidence." Cross–Appellant Br. 35.

5 Oracle also argues that Google conceded that its use was "entirely commercial" during oral argument to this court in the first appeal. *Order Denying JMOL*, 2016 WL 3181206, at *7 ("Q: But for purpose and character, though, you don't dispute that it was entirely a commercial purpose. A: No."). The district court treated this colloquy as a judicial admission that Google's use was "commercial." *Id*. (noting that the word "entirely" was "part of the give and take" of oral argument). The court therefore instructed the jury that Google's use was commercial, but that it was up to the jury to determine the extent of the commerciality. *Id*. at *8. Oracle does not challenge the district court's jury instructions on appeal. In any event, as the district court noted, "even a wholly commercial use may still constitute fair use." *Id*. at *7 (citing *Campbell*, 510 U.S. at 585, 114 S.Ct. 1164).

6 According to the district court, if this fact were sufficient to defeat fair use, "it would be impossible ever to duplicate declaring code as fair use and presumably the Federal Circuit would have disallowed this factor on the first appeal rather than remanding for a jury trial." *Id*. But in our prior decision, we remanded in part because Google represented to this court that there were disputes of fact regarding how Android was used and whether the APIs Google copied served the same function in Android and Java. *Oracle*, 750 F.3d at 1376. Without the benefit of briefs exploring the record on these issues, and Google's later agreement with respect to these facts, we concluded that we could not say that there were no material facts in dispute. *Id*. As explained previously, however, those facts are no longer in dispute. The only question that remains regarding transformative use is whether, on the now undisputed facts, Google's use of the APIs was, in fact, transformative.

7 Because we conclude that smartphones were not a new context, we need not address the argument, made by Oracle and certain amici, that the district court's order excluding evidence of Google's use of Android in multiple other circumstances —including laptops—tainted the jury's and the court's ability to fairly assess the character of the use.

8 As some amici note, to hold otherwise could encroach upon the copyright holder's right to "prepare derivative works based upon the copyrighted work." 17 U.S.C. § 106(2); *see* Br. of Amicus Curiae N.Y. Intell. Prop. L. Ass'n at 17–20.

9 As the district court recognized, there is some debate about whether good or bad faith should remain relevant to the factor one inquiry. *Order Denying JMOL*, 2016 WL 3181206, at *2 ("[T]here is a respectable view that good or bad faith should no longer be a consideration after the Supreme Court's decision in *Campbell*."); *see also* Hon. Pierre N. Leval, *Toward a Fair Use Standard*, 103 Harv. L. Rev. 1105, 1128 (1990) ("Whether the secondary use is within the protection of the [fair use] doctrine depends on factors pertinent to the objectives of the copyright law and not on the morality or motives of either the secondary user or the copyright-owning plaintiff."). In *Campbell*, the Supreme Court expressed skepticism about "the weight one might place on the alleged infringer's state of mind." *Campbell*, 510 U.S. at 585 n.18, 114 S.Ct. 1164. But the Ninth Circuit has not repudiated its view that " 'the propriety of the defendant's conduct' is relevant to the character of the use at least to the extent that it may knowingly have exploited a purloined work for free that could have been obtained for a fee." *L.A. News Serv. v. KCAL–TV Channel 9*, 108 F.3d 1119, 1122 (9th Cir. 1997) (quoting *Harper & Row*, 471 U.S. at 562, 105 S.Ct. 2218). For that reason, and because we conclude in any event that the jury must have found that Google did not act in bad faith, we address that question and the parties' arguments relating thereto.

10 The jury was instructed that, "[i]n evaluating the extent to which Google acted in good faith or not, you may take into account, together with all other circumstances, the extent to which Google relied upon or contravened any recognized practices in the industry concerning reimplementation of API libraries." *Order Denying JMOL*, 2016 WL 3181206, at *3 n.2. Oracle has not challenged this instruction on appeal.

11 In the prior appeal, we noted that "Google's competitive desire to achieve commercial 'interoperability' ... may be relevant to a fair use analysis." *Oracle*, 750 F.3d at 1376–77. But, although several amici in this appeal discuss interoperability concerns, Google has abandoned the arguments it once made about interoperability. This change in course is not surprising given the unrebutted evidence that Google specifically designed Android to be *incompatible* with the Java platform and not allow for interoperability with Java programs. *Id*. at 1371.

12 The Court noted, however, that "what *Sony* said simply makes common sense: when a commercial use amounts to mere duplication of the entirety of an original, it clearly 'supersede[s] the objects,' of the original and serves as a market replacement for it, making it likely that cognizable market harm to the original will occur." *Id*. at 591, 114 S.Ct. 1164.

13 Google submits that the jury could have discounted this evidence because the Java SE APIs were available for free through OpenJDK. But Amazon moved from Java to Android—not to OpenJDK. And the evidence of record makes clear that device manufacturers did not view OpenJDK as a commercially viable alternative to using Java SE because any improvement to the packages in OpenJDK had to be given away for free to the Java community.

14 Of course, the fact that those negotiations were not successful does not factor into the analysis. *Campbell*, 510 U.S. at 585 n.18, 114 S.Ct. 1164 ("If the use is otherwise fair, then no permission need be sought or granted. Thus, being

denied permission to use a work does not weigh against a finding of fair use."). Such evidence was only relevant to show Oracle's interest in the potential market for smartphones.

End of Document

CERTIFICATE OF COMPLIANCE

This petition complies with the type-volume limitation of Federal Rule of Appellate Procedure 35(b)(2) because it contains 3,892 words. This motion also complies with the substantive requirements of Federal Circuit Rule 35(b)–(c).

This petition complies with type-face requirements of Federal Rule of Appellate Procedure 32(a)(5) and the type-style requirements of Federal Rule of Appellate Procedure 32(a)(6). This motion has been prepared in a proportionally spaced typeface using Microsoft Office Word Version 2013 in 14-point Century Schoolbook font.

/s/ *Daryl L. Joseffer*
Daryl L. Joseffer

CERTIFICATE OF SERVICE

In accordance with Federal Rule of Appellate Procedure 25 and Federal Circuit Rule 25, I certify that on May 29, 2018, I caused a copy of the foregoing petition to be served electronically on all registered counsel through the Court's CM/ECF system.

/s/ *Daryl L. Joseffer*
Daryl L. Joseffer