Harvard Journal of Law & Technology
Volume 31, Special Issue Spring 2018

SAVING SOFTWARE'S FAIR USE FUTURE

Pamela Samuelson & Clark D. Asay[*]

TABLE OF CONTENTS

[*] Pamela Samuelson is Richard M. Sherman Distinguished Professor of Law, Berkeley Law. Clark D. Asay is Associate Professor of Law, BYU Law School. We wish to thank Kathryn Hashimoto for research and editorial support. This Article is a derivative work of the Brief of *Amici Curiae* 42 Intellectual Property Law Professors in Support of Defendant-Cross Appellant Google Inc., Oracle Am., Inc. v. Google Inc. Nos. 2017-1118–1202 (Fed. Cir. appeal docketed Oct. 28, 2016). Owing to space considerations, the names of the signatories are not included in this Article but can be found in the brief, available at https://ssrn.com/abstract=2982564.

I. INTRODUCTION

The freedom to reimplement application program interfaces (APIs) in independently written software is the key issue at stake in *Oracle America, Inc. v. Google Inc.*[1] Properly resolving this issue is crucial because APIs play such an important role in modern software innovation. For instance, reuse of APIs is vital to enabling interoperability among distinct software programs.[2] Reuse of APIs also allows software programmers to more rapidly build upon the work of others in developing innovative new software.[3] The issue at the heart of the *Oracle v. Google* case, then, could not be more important for the modern software industry.

Although most cases testing the legality of unlicensed reimplementations of APIs have been decided on copyrightability grounds,[4] the *Oracle* case tests the viability of fair use as a defense to claims of copyright infringement for API reuses. How the Federal Circuit decides Oracle's appeal of a jury verdict in favor of Google's fair use

[1] 872 F. Supp. 2d 974 (N.D. Cal. 2012) (No. C 10-03561 WHA), *rev'd and remanded*, 750 F.3d 1339 (Fed. Cir. 2014), *cert. denied*, 135 S. Ct. 2887 (2015).

[2] *See, e.g.*, Clark D. Asay, *Software's Copyright Anticommons*, 66 EMORY L.J. 265, 286–88 (2017) (discussing the growing interoperability needs in the modern software industry and the importance of API reuse in that context).

[3] *Id.* at 280–85 (discussing the growing collaborative nature of modern software development and the need of software interface reuse in order to enable that collaboration).

[4] *See infra* Section III.B.

defense will have significant implications for future software copyright fair use cases because Oracle, in effect, calls into question the viability of fair use defenses in all API reuse cases (and perhaps in software cases more generally). Fair use in the digital age has come to play an important role in balancing the interests of first- and second-generation creators in software as well as other creative fields. So, it would be not just unfortunate, but possibly devastating to competition and ongoing innovation in the software industry if fair use defenses were precluded in API reuse cases. This Article challenges Oracle's arguments that as a matter of law, no reasonable jury could have upheld Google's fair use defense, while also highlighting important considerations for future courts in the software fair use context.

Before delving into the legal issues, we offer a brief summary of the case. Oracle sued Google in 2010, claiming that the Java API is protectable by copyright law and that Google's use of 37 of the 166 packages of that API in its Android platform without a license was copyright infringement. The District Court initially ruled that the elements of Java replicated by Google were not subject to copyright protection because they were too utilitarian to qualify as copyrightable expression.[5] On appeal in 2014, the Federal Circuit reversed this ruling on the copyrightability issue,[6] and the U.S. Supreme Court denied Google's petition to review that decision.[7] Yet, the Federal Circuit agreed with Google that there were triable issues of fact about whether Google had made fair use of the Java API.[8] On remand, a jury in May 2016 found Google's use of the Java API elements was a fair use.[9] After the District Court declined to vacate the jury verdict,[10] Oracle once again appealed to the Federal Circuit, claiming that it is entitled to a judgment in its favor on the fair use issue as a matter of law.[11]

[5] *Oracle*, 872 F. Supp. 2d at 976–77. One of us has elsewhere argued that the District Court's decision on copyrightability was sound. *See* Pamela Samuelson, *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, 31 BERKELEY TECH. L. J. 1215, 1238–67 (2016).

[6] Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1356–61 (Fed. Cir. 2014). The Federal Circuit ruled that the Java API was original enough to qualify for copyright protection and not disqualified on functionality grounds from copyright because Oracle's choices for names and organizations of the API elements were not dictated by function. *Id.*

[7] Google Inc. v. Oracle Am., Inc., 135 S. Ct. 2887 (2015), *denying cert. to* 750 F.3d 1339 (Fed. Cir. 2014).

[8] *Oracle*, 750 F.3d at 1372–77.

[9] Oracle Am., Inc. v. Google Inc., No. C 10-03561 WHA, 2016 WL 3181206 at *1 (N.D. Cal. June 8, 2016) (order denying Oracle's renewed motion for a judgment as a matter of law and motion for a new trial).

[10] *Id.*

[11] Opening Brief and Addendum for Oracle America, Inc. at 4, Oracle Am., Inc. v. Google Inc., Nos. 2017-1118, -1202 (Fed Cir. Feb. 10, 2017) [hereinafter Oracle Brief]. Oracle has also asked for a new trial, claiming that the District Court erred in not allowing

Oracle's attack on the jury's fair use verdict rests on numerous flawed assertions that, if accepted by the Federal Circuit, could undermine robust software innovation by severely restricting the fair use defense's applicability in software copyright cases. Part II addresses Oracle's erroneous claim that Google's failure to license certain elements of the Java API for its Android platform was evidence of bad faith as a matter of law.[12] This assertion is plainly inconsistent with the Supreme Court's decision in *Campbell v. Acuff-Rose Music, Inc.*,[13] in which the Court not only expressed skepticism about whether subjective "faith" — good or bad — is relevant in fair use cases, but also explicitly stated that seeking, but not obtaining, a license to use another's copyrighted material should not be considered evidence of bad faith. Oracle has also mistakenly asserted that courts and juries may consider only evidence of subjective bad faith, and not evidence of good faith, in assessing fair use.[14] However, numerous decisions have considered defendants' good faith in fair use cases.[15]

Part III rebuts Oracle's argument that Google could not have acted in good faith when reusing parts of the Java API. Google witnesses testified to a common industry belief that reimplementing APIs is permissible.[16] Google could reasonably have relied on this belief, as well as on several appellate court decisions that strictly limited the scope of copyright protection for software interfaces.[17] These rulings are, moreover, consistent with the position of the American Committee for Interoperable Systems (ACIS) that historically supported the freedom to reimplement software interfaces. ACIS, of which both Sun Microsystems (the original developer of the Java API) and Oracle were members, filed numerous briefs in support of the proposition that APIs were not and should not be copyrightable expression.[18]

Part IV challenges, on three grounds, Oracle's contention that the jury could not, as a matter of law, have found Google's purpose in reusing elements of the Java API to be transformative because Google used those elements to perform the same computing functions they were designed to perform.[19] First, this contention is inconsistent with the Federal Circuit's previous ruling that there was a triable issue of

Oracle to present certain evidence at trial. *Id.* at 3–4. Oral argument on Oracle's appeal was scheduled for December 7, 2017.

[12] Oracle Brief, *supra* note 11, at 37–38.

[13] 510 U.S. 569 (1994).

[14] Oracle Brief, *supra* note 11, at 28, 38.

[15] *See infra* Sections II.B and II.C.

[16] *See* Oracle Am., Inc. v. Google Inc., No. C 10-03561 WHA, 2016 WL 3181206, at *2 (N.D. Cal. June 8, 2016); *see also infra* note 54 and accompanying text.

[17] *See infra* Sections III.B.

[18] Copies of those briefs can be found at http://www.ccianet.org/interop/ [https://perma.cc/57G7-P9E9].

[19] Oracle Brief, *supra* note 11, at 29–31.

fact on Google's fair use defense, including on transformativeness.[20] Second, binding Ninth Circuit precedent has treated similar reuses of software interfaces as transformative.[21] Third, other technological fair use decisions lend support to Google's transformativeness argument.[22] Numerous fair use precedents have allowed reuse of copyrighted works, including computer code, to enable technological competition and innovation, including but not limited to reverse engineering, emulation, interoperability, data-mining, image recognition, plagiarism detection, and information location. Accepting Oracle's non-transformative use arguments would, in fact, largely preclude fair use as a defense in the software context more generally, since the utilitarian nature of software means that reusing software, including APIs, will almost always implicate the same computing functions that the software was originally designed to perform.

Part IV shows that Oracle has mischaracterized the second fair use factor by focusing on the creativity it took to develop the Java API.[23] That fair use factor is, however, concerned with the nature of the work at issue, not the degree of creativity required to develop an API. Under binding Ninth Circuit precedent, computer programs are deemed utilitarian works.[24] Such works enjoy a thin scope of copyright protection and are subject to a broad scope of fair use, no matter how much creativity it took to develop them. The jury's fair use verdict in *Oracle* strongly suggests that the jurors were persuaded that the Java API elements Google used in Android were more functional than expressive. If the Federal Circuit adopts Oracle's viewpoint on the nature-of-the-work factor, that would likely undercut fair use defenses in software copyright cases generally, not just in reuse of API cases.

II. BAD OR GOOD FAITH HAS LITTLE RELEVANCE IN FAIR USE CASES

The Federal Circuit has yet to express an opinion on whether showings of good or bad faith should help or hurt fair use defenses. Oracle hopes to persuade the Federal Circuit to adopt its view that bad faith precludes fair use. Its "most emphatic argument" in the District Court, and one of its most aggressive arguments on appeal, was that Google acted in bad faith, as a matter of law, by not taking a license

[20] Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1376 (Fed. Cir. 2014).

[21] *See infra* Section IV.A.

[22] *See infra* Section IV.B.

[23] Oracle Brief, *supra* note 11, at 39–43.

[24] *See, e.g.*, Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1442–43 (9th Cir. 1994); Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1524–25 (9th Cir. 1992).

for its use of portions of the Java API in Android.[25] In its appeal brief, Oracle characterized bad faith as "a one-way ratchet: Bad faith weighs against fair use, while a copyist's good faith cannot weigh in favor of fair use."[26] The Federal Circuit should reject Oracle's arguments because these assertions are plainly inconsistent with the Supreme Court's important fair use ruling in *Campbell v. Acuff-Rose Music, Inc.*[27] If accepted, Oracle's position would threaten the role of fair use in fostering productive competition in the software industry.

A. Subjective Mental States Should Be Given Little, If Any, Weight in Fair Use Cases

In *Campbell*, the Supreme Court expressed some skepticism about whether good or bad faith should be given weight in fair use cases.[28] The Court's caution about this consideration may explain why the Federal Circuit, when remanding the *Oracle* case for trial on Google's fair use defense, did not highlight the issue as a relevant consideration for the fair use trial.[29]

Commentators have offered several reasons why mental states, such as subjective good and bad faith, should generally be given relatively little, if any, weight in fair use cases. In his influential article on fair use, Judge Leval observed:

> 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.[30]

Professor Dratler has offered both statutory and economic policy rationales for limiting moral considerations such as good or bad faith in fair use cases:

> First, from the standpoint of faithfulness to statutory language, a user's course of dealing with the holder

[25] Oracle Am., Inc. v. Google Inc., No. C 10-03561 WHA, 2016 WL 3181206, at *2 (N.D. Cal. June 8, 2016); Oracle Brief, *supra* note 11, at 28, 37–39.

[26] Oracle Brief, *supra* note 11, at 28.

[27] 510 U.S. 569 (1994).

[28] *Id.* at 585 n.18 ("*Even if good faith were central to fair use*, 2 Live Crew's actions do not necessarily suggest that they believed their version was not fair use . . . .") (emphasis added).

[29] Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1372–76 (Fed. Cir. 2014); *Oracle*, 2016 WL 3181206, at *2.

[30] Hon. Pierre N. Leval, *Toward a Fair Use Standard*, 103 HARV. L. REV. 1105, 1128 (1990).

> of copyright in the underlying work has little relation to the "purpose" of the use. Second, and more important, there is little reason to infuse the doctrine of fair use with notions of commercial ethics.
>
> Unlike the doctrine of trade secrecy, the doctrine of fair use has no substantial basis in commercial morality. Like copyright law generally, fair use has an economic purpose. The morality *vel non* of transactions between users and copyright holders has little to do with that purpose. Indeed, the very term "fair use" is a misnomer because the doctrine, as codified today, does not focus on notions of ethics and fairness, but on market impacts and the relative public benefits of use versus incentives for creation.[31]

In cases such as *Oracle*, these statutory and economic considerations are especially pertinent because fair use has become an important means of balancing competition and ongoing innovation policy considerations in such cases.[32] Oracle's effort to cast this case in good-guy/bad-guy terms diverts attention away from the underlying economic principles of copyright, which aim to provide a reasonable degree of protection for copyrightable expression, while leaving room for second-comers to build upon preexisting works.

In *Campbell*, the Supreme Court did not altogether abjure consideration of good or bad faith in fair use cases. Yet it rejected Acuff-Rose's argument that Campbell's bad faith was a consideration that undercut his fair use defense. The Court said that "*[e]ven if good faith were central to fair use*, 2 Live Crew's actions do not necessarily suggest that they believed their version was not fair use."[33] This cautionary statement, in keeping with the policy arguments noted above, suggests that subjective mental states such as good/bad faith should generally be given little weight in fair use cases. Yet, insofar as good/bad faith has some bearing on whether a use is fair, Oracle's assertions about the role of bad and good faith in fair use analysis are nevertheless decidedly erroneous.

[31] Jay Dratler, Jr., *Distilling the Witches' Brew of Fair Use in Copyright Law*, 43 U. MIAMI L. REV. 233, 334 (1988) (footnotes omitted).

[32] *See, e.g.*, Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1524–27 (9th Cir. 1992); *see also* Pamela Samuelson, *Unbundling Fair Uses*, 77 FORDHAM L. REV. 2537, 2605–10 (2009); Fred von Lohmann, *Fair Use as Innovation Policy*, 23 BERKELEY TECH. L.J. 829 (2008).

[33] *Campbell*, 510 U.S. at 585 n.18 (emphasis added).

B. Seeking, but not Getting, a License Is not Evidence of Bad Faith

Campbell is one of many fair use cases in which putative fair users had, prior to litigation, sought but did not obtain a license for the challenged use of copyrighted materials. Acuff-Rose in *Campbell*, for instance, pointed to 2 Live Crew's request for permission to use Roy Orbison's song "Oh, Pretty Woman" as evidence of bad faith. The Court expressly rejected this argument, saying "2 Live Crew's actions do not necessarily suggest that they believed their version was not fair use; the offer may simply have been made in a good-faith effort to avoid this litigation. *If the use is otherwise fair, then no permission need be sought or granted*."[34] The Court cited approvingly to *Fisher v. Dees*, in which the Ninth Circuit observed that "to consider Dees blameworthy because he asked permission would penalize him for this modest show of consideration."[35] Thus, Oracle's argument that Google acted in bad faith by seeking, but not obtaining, a license to use the Java API is flatly inconsistent with the Supreme Court's decision in *Campbell*.[36]

Campbell and *Dees* are far from the only cases in which fair use defenses prevailed even though the parties had failed to agree on licensing terms.[37] Perhaps most pertinent to the *Oracle* case is the Ninth Circuit's decision in *Sega Enters. v. Accolade, Inc.*[38] Prior to engaging in the reverse-engineering activity that precipitated the lawsuit, Accolade approached Sega about taking a license to make videogames for the Genesis platform.[39] Accolade decided against accepting the license that was offered because it found unacceptable one of Sega's key licensing terms: a commitment to make games only for the Sega platform.[40] The failed negotiations did not undermine Acco-

[34] *Id.* (emphasis added).

[35] Fisher v. Dees, 794 F.2d 432, 437 (9th Cir. 1986), *cited in Campbell*, 510 U.S. at 585 n.18 ("[t]hus, being denied permission to use a work does not weigh against a finding of fair use.").

[36] Oracle Brief, *supra* note 11, at 11–12.

[37] *See, e.g.*, Maxtone-Graham v. Burtchaell, 803 F.2d 1253, 1264 (2d Cir. 1986) (quoting from plaintiff's work after being denied permission was not evidence of bad faith, for the defendant "should not be penalized for erring on the side of safety"); Warren Publ'g Co. v. Spurlock, 645 F. Supp. 2d 402, 422 (E.D. Pa. 2009) (following *Campbell* in rejecting plaintiff's assertion that defendant's failed attempts to negotiate a license to reproduce magazine covers as part of artist's retrospective amounted to bad faith); Equals Three, LLC v. Jukin Media, Inc., 139 F. Supp. 3d 1094, 1106 (C.D. Cal. 2015) ("If using a song after requesting and being denied a license does not show bad faith, then neither does failing to obtain a license and continuing to use footage after being sent a demand letter." (citing *Campbell*, 510 U.S. at 585 n.18)); Threshold Media Corp. v. Relativity Media, LLC, 166 F. Supp. 3d 1011, 1029 n.13 (C.D. Cal. 2013) (finding it "irrelevant" whether defendants sought permission to use plaintiff's song (citing *Campbell*, 510 U.S. at 585 n.18)).

[38] 977 F.2d 1510, 1514 (9th Cir. 1992).

[39] *Id.* at 1514.

[40] *Id.*

lade's fair use defense, which the Ninth Circuit found quite compelling.[41]

Restricting the fair use defense based on a party's failure to ultimately obtain a license would be a particularly pernicious development in the modern software industry, which is increasingly characterized by norms of openness and collaborative development.[42] Such a turn in the law could contribute to upsetting those norms, thereby threatening the collaborative software ecosystem that they have enabled.

C. Good Faith Can Cut in Favor of Fair Use

Also contrary to Oracle's assertions are precedents treating the good faith of a putative fair user as weighing in favor of fair use. There are even cases that have regarded seeking, but not obtaining, a license as evidence of good faith in support of fair use. In *Bill Graham Archives v. Dorling Kindersley Ltd.*,[43] for instance, the publisher of a book on the cultural history of the Grateful Dead approached BGA about a license to reproduce certain posters that had advertised the band's performances.[44] After the parties failed to reach agreement, the publisher used the images anyway, and BGA sued for infringement.[45] The District Court noted that the defendants had "informed plaintiff of their intentions to use their images and made an effort to license the images where there might be question as to whether a license was needed, [which] shows a good-faith effort by defendants."[46]

Dorling Kindersley is one of many cases that have given favorable consideration to a defendant's good-faith conduct. Another example is *Wright v. Warner Books, Inc.*,[47] involving a biographer's use of an author's journal entries and letters.[48] The Second Circuit favorably described defendant's conduct — which included rightful possession of the letters and offering those letters to the plaintiff before publication of the biography — as refuting plaintiff's bad faith claims.[49]

[41] *See id.* at 1527–28.

[42] *See, e.g.*, Clark D. Asay, *A Case for the Public Domain*, 74 OHIO ST. L.J. 753, 765–68 (2013) (describing these norms and the successes they have yielded).

[43] 386 F. Supp. 2d 324 (S.D.N.Y. 2005), *aff'd*, 448 F.3d 605 (2d Cir. 2006).

[44] 386 F. Supp. 2d at 326.

[45] *Id.*

[46] *Id.* at 333; *see also* Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 585 n.18 (1994) ("[T]he offer [to license] may simply have been made in a good-faith effort to avoid this litigation . . . .").

[47] 953 F.2d 731 (2d Cir. 1991).

[48] *Id.* at 737.

[49] *Id.* at 737; *see also* Nunez v. Caribbean Int'l News Corp., 235 F.3d 18, 23 (1st Cir. 2000) (finding good faith when newspaper acknowledged the photographer and believed that the photos were available for general unrestricted circulation). The Goldstein treatise notes that bad and good faith have been considered relevant in certain cases. *See* 2 PAUL

Within the Ninth Circuit, another court viewed Google's compliance with industry-standard protocols and conduct regarding its cached links to web pages as an indication of its good faith.[50] Oracle's assertion that a defendant's good faith can never weigh in favor of fair use is thus plainly mistaken.

III. UNLICENSED REUSES OF APIS MAY BE DONE IN GOOD FAITH

On appeal to the Federal Circuit, Oracle seems to have taken the position that it is always bad faith not to license reuses of APIs.[51] This contravenes a longstanding norm in the software industry that reuse of APIs in independently written software is lawful, as well as holdings in significant software copyright cases and, indeed, positions that Sun and Oracle have taken on reuses of APIs as members of ACIS.

If the Federal Circuit accepts Oracle's radical position on this issue, it would represent a dramatic departure from how courts and the software industry (including Oracle and Sun, the Java API's original developer) have traditionally perceived the scope of software copyright. And upsetting those settled expectations could severely threaten ongoing robust software innovation. Reuse of APIs enables interoperability among programs, which in turn, promotes competition in the

GOLDSTEIN, GOLDSTEIN ON COPYRIGHT § 12.2.2, at 12:44.5 (3d ed. 2014 & Supp. 2017) ("Courts have also weighed the defendant's bad faith against a fair use finding and the defendant's good faith in favor of a fair use finding." (citing cases)). That a good-faith belief in the fairness of a use may weigh in favor of an alleged infringer is also supported in the Copyright Act, which directs courts to remit statutory damages "in any case" involving certain nonprofit actors who reasonably believed their public-benefiting uses were fair. 17 U.S.C. § 504(c)(2) (2012).

[50] *See* Field v. Google Inc., 412 F. Supp. 2d 1106, 1122–23 (D. Nev. 2006); *see also* Fuentes v. Mega Media Holdings, Inc., No. 09-22979-CIV, 2011 WL 2601356 at *11–13 (S.D. Fla. June 9, 2011) (finding that the record supported defendant news program's "good faith position" regarding its acquisition of video footage); Kane v. Comedy Partners, No. 00 Civ. 158(GBD), 2003 WL 22383387 at *7 (S.D.N.Y. Oct. 16, 2003) (finding it an act of good faith for defendant to notify the plaintiff of its use of a clip of its video); Time, Inc. v. Bernard Geis Assocs., 293 F. Supp. 130, 146 (S.D.N.Y. 1968) (describing as a "strong point" that defendants offered to transfer all profits from sales of the book as royalty payment to use Zapruder frames). Courts also have found a defendant's demonstrated belief that its use was fair can itself be an indication of good faith. *See, e.g.*, Bouchat v. Baltimore Ravens Ltd. P'ship, 737 F.3d 932, 942 (4th Cir. 2013) (declining to disturb the district court's finding that "there is nothing to put into doubt the [defendant]'s good faith in believing that the uses . . . were non-infringing fair uses."); Gulfstream Aerospace Corp. v. Camp Sys. Int'l, Inc., 428 F. Supp. 2d 1369, 1378 n.5 (S.D. Ga. 2006) ("Any bad faith inference that might be drawn is negated by other evidence demonstrating [defendant]'s good faith belief that it was using the manuals appropriately.").

[51] This is akin to the position that the Sixth Circuit took in *Bridgeport Music, Inc. v. Dimension Films*, 410 F.3d 792 (6th Cir. 2005), which insisted that licenses were necessary to lawfully engage in sampling of sound recordings. *Id.* at 801. The Ninth Circuit recently criticized the *Bridgeport* ruling in *VMG Salsoul, LLC v. Ciccone*. 824 F.3d 871, 886–87 (9th Cir. 2016).

software industry and enables ongoing innovation.[52] Fair use should remain one of the doctrines through which these considerations may come into play.

A. A Common Understanding Exists in the Software Industry That APIs Do Not Need to Be Licensed

Like many software developers who have reused other firms' APIs,[53] Google could have had a good-faith belief that its use of the Java API elements would not infringe Oracle copyrights, based on a common understanding in the software industry that interfaces are open for reuse.[54] Testimony in the *Oracle v. Google* fair use trial record, including by Sun's last CEO, supported Google's contention that there was a common understanding in the software industry that programmers were free to reimplement APIs in independently written code. As the District Court noted:

> [Google's] witnesses testified that they had understood that "re-implementing" an API library was a legitimate, recognized practice so long as all that was duplicated was the "declaring code" and so long as the duplicator supplied its own "implementing code," that is, the methods were "re-implemented." In this way, Java programmers using the Android API could call on functionalities with the same Java command statements needed to call the same functionalities in the Java API, thereby avoiding splintering of the ways that identical functionalities became invoked by Java programmers.[55]

[52] *See, e.g.*, Joseph Gratz & Mark A. Lemley, *Platforms and Interoperability in* Oracle v. Google, 31 HARV. J.L. & TECH. (SPECIAL ISSUE) 603 (2018). For an international perspective on API copyright issues, *see* Jonathan Band, *The Global API Copyright Conflict*, 31 HARV. J.L. & TECH. (SPECIAL ISSUE) 615 (2018).

[53] *See infra* Section III.B for a discussion of cases in which re-users of APIs have prevailed in software copyright cases.

[54] To the extent that good/bad faith issues are relevant in fair use cases such as *Oracle*, it is important to recognize, as the District Court did in this case, first, that a plaintiff's insistence that bad faith should be part of the case opened up the opportunity for the defendant to offer evidence that it had acted in good faith. *See* Oracle Am., Inc. v. Google Inc., 2016 WL 3181206, at *2–3 (N.D. Cal. June 8, 2016) (No. C 10-03561 WHA). Second, when the evidence about bad/good faith is mixed, resolution of such a factual dispute is a classic issue of fact for a jury to decide. *Id.* at *6–7. Third, when a license about which the parties had been negotiating was for a wider use of protected works, such as the full range of the Java technologies including the implementing code, a defendant such as Google could have reasonably believed that using only some of the Java declarations and classes was lawful. *Id.* at *11.

[55] *Id.* at *2.

The distinction between interfaces and implementations is longstanding in the computing field.[56]

APIs are often published so that follow-on software developers can refer to them when making decisions about whether to create software for a particular platform. This is especially true as to the Java API. The Java Standard Edition (SE) document, which sets forth the Java API, is available for free downloading on Oracle's website.[57] In addition, a large number of books have been published that set forth the Java API, in whole or in part, including the declarations and class structures.[58] These books generally aim to explain how to use the Java API in an effective manner to develop new software programs. They often reproduce the whole or substantial parts of the Java API for commercial purposes and could serve as substitutes for the Java SE.[59]

The District Court took note of these books in denying Oracle's motion for a judgment as a matter of law after Google won its fair use defense:

> Many thousands of pre-written methods have been written for Java, so many that thick books . . . are needed to explain them, organized by packages, classes, and methods. For each method, the book sets forth the precise declaring code but does not (and need not) set forth any implementing code. In other words, the book duplicates all of the method declarations (organized by packages and classes) together with plain English explanations. A Java user can study the book and learn the exact method name and inputs needed to invoke a method for use in his or her own program . . . . [A]ll that the Java programmer need master are the declarations. The imple-

[56] *See, e.g.*, Alfred Z. Spector, *Software, Interface, and Implementation*, 30 JURIMETRICS J. 79, 85–86 (1989).

[57] *See* Oracle JDK 9 Documentation, https://docs.oracle.com/javase/9/ [https://perma.cc/T53B-SPYE].

[58] For example, Amazon offers a Java Books category of products with more than 5,000 results. *Java Programming*, AMAZON, https://www.amazon.com/Java-Programming-Computers-Internet-Books/b?ie=UTF8&node=3608 [https://perma.cc/BQ2N-H95X]. One such book is IAN F. DARWIN, JAVA COOKBOOK: SOLUTIONS AND EXAMPLES FOR JAVA DEVELOPERS (3d ed. 2014).

[59] *See, e.g.*, DARWIN, *supra* note 58, at 767–70 ("Finding and Using Methods and Fields"). Google's use of the Java API elements was, moreover, not for explanatory purposes, which could implicate copyright, but for functional purposes, which copyright does not generally regulate. *See* Wendy J. Gordon, *How* Oracle *Erred: The Use/Explanation Distinction and the Future of Computer Copyright* in COPYRIGHT LAW IN AN AGE OF LIMITATIONS AND EXCEPTIONS (Ruth Okediji, ed., 2017).

> menting code remains a "black box" to the programmer.[60]

In view of the Google witness testimony about the common understanding of their freedom to reimplement APIs and common uses of the Java API declarations in books, the jury could reasonably have decided that Google acted in good faith in reusing elements of the Java API in Android. Similar considerations will likely be at play in future cases involving reuses of APIs. Future courts, contrary to Oracle's arguments, should thus view such considerations as evidence of good faith in favor of fair use.

B. Numerous Precedents Have Rejected Expansive Copyright Claims in Software APIs

When deciding whether to reuse another firm's APIs without a license, software developers such as Google have generally been able to rely on several appellate court cases, including two significant Ninth Circuit precedents, that rejected copyright claims pertaining to computer program interfaces. Although the doctrinal rationales enabling unlicensed firms to reuse program APIs have sometimes varied, the software copyright case law has generally upheld API reuse defenses, at least until the Federal Circuit's *Oracle* copyrightability ruling. One of the most worrisome aspects of the *Oracle* decision was that its reasoning about the copyrightability of the Java API elements used by Google called into question the holdings in other interoperability cases.[61] Consider the interoperability defenses in the following six contexts.[62]

One context was in *Sega Enters. v. Accolade, Inc.*[63] There, an unlicensed game developer sought to make its games run on Sega's popular platform.[64] Sega had developed an interface to enable videogames to run on its Genesis platform.[65] Once that interface existed, Sega and its licensees had to conform to the interface specifications when developing games for the Genesis.[66] The only way that

[60] *Oracle*, 2016 WL 3181206, at *4.

[61] *See, e.g.*, Pamela Samuelson, *Three Fundamental Flaws of the CAFC's* Oracle v. Google *Decision*, 37 EUR. INTELL. PROP. REV. 702 (Nov. 2015). The *Oracle* decision has emboldened other software developers to sue competitors for reuses of APIs and command structures. *See, e.g.*, Fred von Lohmann, *The New Wave: Copyright and Software Interfaces in the Wake of* Oracle v. Google, 31 HARV. J.L. & TECH. (SPECIAL EDITION) 517 (2018).

[62] For a more extensive discussion of the interface cases, see Samuelson, *Functionality and Expression*, *supra* note 5, at 1245–67.

[63] 977 F.2d 1510 (9th Cir. 1992).

[64] *Id.* at 1514–15.

[65] *Id.*

[66] *Id.*

Accolade could make its games work on the Sega platform was by reimplementing the Sega interface details in its game software.[67] The Ninth Circuit characterized the Sega interface as the "functional requirements for compatibility," which were among the "procedures" excluded from copyright protection under 17 U.S.C. § 102(b).[68]

A second context was in *Sony Computer Entertainment v. Connectix Corp.*[69] Sony, maker of the popular PlayStation platform, sued the developer of software that emulated the platform's functionality.[70] Connectix did so to attract owners of PlayStation games to play them using its software.[71] To accomplish this objective, Connectix first studied the BIOS of the Sony PlayStation, then reimplemented the BIOS interface in independently written code.[72] In keeping with its *Accolade* decision, the Ninth Circuit characterized the program interface as an unprotectable procedure under 17 U.S.C. § 102(b).[73]

A third context was in *Computer Associates International, Inc. v. Altai, Inc.*[74] Both litigants were competitors in the market for scheduling programs designed to run on certain IBM operating system programs.[75] Computer Associates International (CAI) argued that Altai infringed by copying the structure, sequence, and organization of its list of services as well as parameter lists that set forth interfaces for interacting with the IBM programs.[76] The Second Circuit rejected CAI's argument, finding the list of services "was dictated by the nature of other programs with which it was designed to interact" and the parameter lists were not similar enough to infringe.[77] The IBM programs had constrained the design choices for both CAI and Altai in the formulation of parameter lists.[78]

[67] *Id.*

[68] *Id.* at 1522. Section 102(b) states: "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." For a critique of the Federal Circuit's interpretation of this provision, see Samuelson, *Functionality and Expression, supra* note 5, at 1237–67. The Ninth Circuit's decision in *Bikram's Yoga College of India, L.P. v. Evolation Yoga, LLC*, 803 F.3d 1032 (9th Cir. 2015), which ruled that Bikram's sequence of yoga positions was an unprotectable system under § 102(b), undercuts the Federal Circuit's claim that its interpretation of § 102(b) follows Ninth Circuit precedents.

[69] 203 F.3d 596 (9th Cir. 2000).

[70] *Id.* at 598.

[71] *Id.*

[72] *Id.*

[73] *Id.* at 602–03.

[74] 982 F.2d 693 (2d Cir. 1992).

[75] *Id.* at 699–700.

[76] *Id.* at 714–15.

[77] *Id.* at 715.

[78] *Id.*

A fourth context was in *Bateman v. Mnemonics, Inc*.[79] Mnemonics developed an application program to run on an operating system program it licensed from Bateman.[80] When Bateman terminated the license, Mnemonics decided to develop an operating system so that it could continue to run its application program.[81] Bateman sued for copyright infringement. The district court disallowed jury consideration of Mnemonics' evidence and argument that some literal copying of Bateman's code was necessary so that its new operating system could execute its application program.[82] The Eleventh Circuit reversed, holding that Mnemonics should be able to make that showing and argument.[83]

A fifth context was in *Lexmark International, Inc. v. Static Control Components, Inc.*[84] which involved a claim of copyright in code that served as an interface between printers and printer cartridges. Lexmark embedded a program in its printer cartridges designed to exchange information with another program embedded in Lexmark printers.[85] This exchange was necessary to authenticate the cartridge and enable it to work in Lexmark's printers. Static developed chips loaded with a copy of the Lexmark cartridge software and sold these chips to Lexmark's competitors who wanted their printer cartridges to work in Lexmark printers.[86] Lexmark sued Static Control for copyright infringement. The Sixth Circuit rejected that claim because any expression in that program had merged with its functionality.[87]

A sixth context was in *Lotus Development Corp. v. Borland International, Inc.*[88] Borland literally copied the user interface command hierarchy of Lotus's popular 1-2-3 spreadsheet program (that is, the selection and arrangement of specific commands for invoking specific functions) for the emulation mode of its competing spreadsheet program.[89] Borland did so to enable prospective customers who had created macro programs for commonly executed sequences of functions in 1-2-3 to run those macros in the Borland program. The First Circuit reversed a district court ruling that copying the command hierarchy was infringement.[90] To enable macro compatibility, the commands

[79] 79 F.3d 1532 (11th Cir. 1996).

[80] *Id.* at 1537–38.

[81] *Id.* at 1540.

[82] *Id.* at 1539, 1543–46.

[83] *Id.*

[84] 387 F.3d 522 (6th Cir. 2004).

[85] *Id.* at 530.

[86] *Id.* at 529–31.

[87] *Id.* at 537–42.

[88] 49 F.3d 807 (1st Cir. 1995).

[89] *Id.* at 810.

[90] *Id.* at 819.

had to be exactly the same and exactly in the same sequence.[91] The First Circuit ruled that the command hierarchy was an uncopyrightable method of operation under 17 U.S.C. § 102(b).[92]

The *Borland* decision also recognized that third-party investments in learning a particular command structure should be a factor cutting against a claim of software copyright infringement.[93] As Judge Boudin noted in his concurrence:

> [I]t is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an investment in learning made by the users and not by Lotus. Lotus has already reaped a substantial reward for being first; assuming that the Borland program is now better, good reasons exist for freeing it to attract old Lotus customers: to enable the old customers to take advantage of a new advance, and to reward Borland in turn for making a better product. If Borland has not made a better product, then customers will remain with Lotus anyway.[94]

Judge Boudin considered fair use as another plausible basis for ruling in Borland's favor.[95] If Google manages to preserve its fair use win before the Federal Circuit, the doctrine of fair use may become another viable defense in cases challenging reuses of APIs as copyright infringement.[96]

In ruling on Oracle's first appeal, the Federal Circuit regarded the *Sega* and *Altai* decisions as distinguishable because in them, the defendant's reuse of the plaintiff's interface was necessary to enable the second-comer software to achieve interoperability with other software.[97] Google, by contrast, used API elements that did not enable such interoperability.[98] While the Federal Circuit has decided that the Java API elements Google used in Android were copyrightable, it also

[91] *Id.* at 818.

[92] *Id.* at 815–19.

[93] *Id.* at 817–18.

[94] *Id.* at 821 (Boudin, J., concurring).

[95] *Id.* at 821–22.

[96] One advantage of the § 102(b) and merger defenses in reuse of API cases over fair use is that lawsuits can generally be decided on motions for summary judgment without incurring the expense of going to trial. For individual software developers, startups, and small firms, the costs of going to trial are generally prohibitive.

[97] Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1368–72 (Fed. Cir. 2014).

[98] *Id.* at 1371.

acknowledged that the compatibility issues that Google had raised might be pertinent to Google's fair use defense.[99]

Also possibly relevant to that defense in *Oracle* were the very substantial investments that millions of Java programmers worldwide had made in learning the Java API declarations and classes as tools with which to write programs in Java.[100] If Oracle's appeal succeeds in overriding the jury's verdict of fair use, millions of programmers may have to learn a whole new dialect of Java in which to write programs for the Android platform. These programmers might also have to rewrite programs previously constructed for the Android platform to use the alternative Java dialect that the lawsuit aims to require Google to develop. A jury might well consider this kind of wasteful expenditure of effort as yet another factor supporting Google's fair use defense.

C. As Members of ACIS, Sun and Oracle Have Supported the Uncopyrightability of APIs

ACIS was formed to promote balanced intellectual property rules for computer software. Among its stated principles to realize this goal was a commitment to the norm that software APIs should not be protected by copyright law, and that independent implementations of APIs should not be considered to infringe copyright. The ACIS Statement of Principles declared: "The rules or specifications according to which data must be organized in order to communicate with another program or computer, i.e., interfaces and access protocols, are not protectable expression under copyright law."[101] Sun was a founding member of ACIS, and Oracle joined ACIS in 1992.[102]

ACIS filed numerous *amicus curiae* briefs supporting freedom to reimplement program interfaces. Sun's Deputy General Counsel, Peter M.C. Choy, was a lead lawyer on numerous *amicus curiae* briefs on behalf of ACIS in major software copyright cases.[103]

[99] *Id.* at 1368.

[100] Oracle Am., Inc. v. Google Inc., No. C 10-03561 WHA, 2016 WL 3181206, at *5–6 (N.D. Cal. June 8, 2016). *See also* Samuelson, *Functionality and Expression*, *supra* note 5, at 1264–65.

[101] *See* AMERICAN COMMITTEE FOR INTEROPERABLE SYSTEMS (ACIS), STATEMENT OF PRINCIPLES, appended to Letter from Peter M.C. Choy on behalf of ACIS to Barry E. Carter (Nov. 5, 1992), *available at* https://www.ccianet.org/wp-content/uploads/2014/10/ACIS-Letter-to-Clinton-Admin-1992.pdf [https://perma.cc/3Q45-G872].

[102] Brief Amicus Curiae of American Committee for Interoperable Systems at iii, Eng'g Dynamics, Inc. v. Structural Software, Inc., 46 F.3d 408 (5th Cir. 1995) (No. 92-3444) (listing ACIS members).

[103] These briefs can be found at http://www.ccianet.org/interop/ [https://perma.cc/57G7-P9E9]. *See, e.g.*, Brief Amicus Curiae of the American Committee for Interoperable Systems, Sega Enters. Ltd. v. Accolade, Inc., 785 F. Supp. 1392 (N.D. Cal. 1992) (Civ No.

Consider this excerpt from the ACIS brief to the Supreme Court in support of Borland's argument in the *Borland* case:

> Unlike traditional literary works such as novels and plays that stand alone and do not need to interact with any other work, computer programs never function alone; they function only by interacting with the computer environment in which their developers place them. This environment is absolutely unforgiving. Unless the computer program conforms to the precise rules for interacting with the other elements of the system, no interaction between the program and the system is possible. As a consequence, no matter how much better or cheaper the new program is, it will not enjoy a single sale if it cannot interoperate in its intended environment. If the developer of one part of the environment can use copyright law to prevent other developers from writing programs that conform to the system of rules governing interaction within the environment — interface specifications, in computer parlance — the first developer could gain a patent-like monopoly over the system without ever subjecting it to the rigorous scrutiny of a patent examination.[104]

The ACIS brief echoes arguments that Google made in the *Oracle* case.[105]

Hence, accepting Oracle's arguments against fair use would upset a consistent position in both the case law and software industry that reusing software interfaces does not constitute copyright infringement. The balance between copyright holders and follow-on innovators that that position reflects has contributed significantly to robust software innovation over the years. Contrary to Oracle's arguments, fair use is and should remain an important part of striking that productive balance.

91-3871-BAC), https://www.ccianet.org/wp-content/uploads/2014/10/Sega-Enterprises-Ltd.-v.-Accolade-Inc.-ACIS-N.D.-Cal..pdf [https://perma.cc/PT8V-P69E].

[104] Brief *Amici Curiae* of American Committee for Interoperable Systems and Computer & Communications Industry Ass'n in Support of Respondent, Lotus Dev. Corp. v. Borland Int'l, Inc., 516 U.S. 233 (1996) (No. 94-2003), 1995 WL 728487 at *4–5.

[105] *See, e.g.*, Brief of Appellee and Cross-Appellant Google Inc. at 1–2, Oracle Am., Inc. v. Google Inc., 750 F.3d 1339 (Fed. Cir. 2014) (ECF Nos. 2013-1021–22).

IV. REUSES OF SOFTWARE APIS CAN BE TRANSFORMATIVE

In the aftermath of the *Campbell* decision, an important consideration in fair use cases has become whether a second comer's use of a first author's work is "transformative."[106] According to *Campbell*, a use may be transformative depending on "whether the new work merely 'supersede[s] the objects' of the original creation, or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message."[107]

In its appeal brief on the fair use issue, Oracle has contended that Google's use of the Java declarations and classes cannot be transformative as a matter of law because Google reused these elements of the Java API for the same functional purpose as the original.[108] This argument is inconsistent with the Federal Circuit's remand of the case for trial on Google's fair use defense because the Federal Circuit said there was a triable issue of fact about whether Google's use of the Java declarations and classes was transformative.[109] The jury's verdict suggests that it was persuaded by Google's arguments on the transformativeness issue. Oracle's argument is also inconsistent with Ninth Circuit software copyright case law and other technology-related fair use rulings. To preserve opportunities for future reusers of APIs to raise fair use defenses, the Federal Circuit should reject Oracle's flawed argument that the reuse of an API cannot be transformative as a matter of law.

A. Reimplementing an API in Independently Written Code Has Been Held to Be Transformative for Fair Use Purposes

Connectix is an important Ninth Circuit precedent recognizing that a defendant's reuse of a software API for the same functional purpose as the plaintiff's use may be transformative.[110] Connectix reverse engineered the PlayStation's BIOS to discover and then extract information about the interface procedures by which the Sony software functioned.[111] Connectix then reimplemented that functionality in its independently written "PlayStation emulator" software.[112]

[106] Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 579 (1994) (holding that transformative uses favor fair use).

[107] *Id.* at 579 (citations omitted). One of us has discussed the various meanings of "transformativeness" in the post-*Campbell* case law. *See* Pamela Samuelson, *Possible Futures of Fair Use*, 90 WASH. L. REV. 815, 825–36 (2015).

[108] Oracle Brief, *supra* note 11, at 29–37.

[109] Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1374–77 (Fed. Cir. 2014).

[110] Sony Computer Entm't v. Connectix Corp., 203 F.3d at 596, 601–02 (9th Cir. 2000).

[111] *Id.*

[112] *Id.*

Even though Connectix's software replicated many of the same computing functions as the PlayStation firmware, the Court held that Connectix's use of the PlayStation firmware was transformative for at least two reasons. First, Connectix's emulator software enabled consumers to use PlayStation games in a new environment (i.e., on personal computers).[113] Second, the Court considered the emulator software "a wholly new product, notwithstanding the similarity of uses and functions" between the PlayStation gaming console and the emulator program, because Connectix had created its own code for implementing the PlayStation firmware's functions in the emulator software.[114]

Google's use of the Java declarations and classes is similar in key respects to Connectix's use of Sony's PlayStation firmware. Both Google and Connectix reimplemented the functionalities of another firm's software in their own software products. Important for purposes of the transformative use inquiry, both Connectix and Google reimplemented those functionalities in new computing environments and wrote their own software code for carrying out those functionalities. Moreover, because of this reimplementation in a wholly new, innovative product, Google's uses of the Java API did not supersede Sun's or Oracle's use of the Java API, which *Campbell* identified as a criterion for non-transformative uses.[115] Instead, Google's use of the Java API in Android added "something new, with a further purpose or different character, altering the first with new expression, meaning, or message."[116]

Accolade is another Ninth Circuit fair use decision that allowed for reuse of software APIs. The Ninth Circuit accepted that Accolade's reimplementation of the Sega Genesis gaming console's interface procedures in its own game products was lawful.[117] As in *Connectix*, Accolade reimplemented the Sega interface procedures in its own software so that its games would function properly on the Genesis console.[118] The Ninth Circuit did not explicitly address the transformativeness issue in *Accolade* because the Supreme Court had yet to endorse that term. But the overall fair use holding supports the proposition that reuses of software interfaces to perform the same basic computing functions can pass fair use muster. As in *Connectix*, the Accolade games were independently created, provided consumers with more choices on a wider array of platforms, and did not super-

[113] *Id.* at 606–07.
[114] *Id.*
[115] Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 578–79 (1994).
[116] *Id.*
[117] Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1527 (9th Cir. 1992).
[118] *Id.* at 1514–17.

sede the market for the original.[119] Given these factors, *Accolade* should be understood as supporting Google's transformativeness arguments.

Connectix and *Accolade* demonstrate that reuse of software APIs can be transformative even when a defendant uses the API to achieve the same computing purpose as the plaintiff and even when the defendant's software is competitive with the plaintiff's work. The Federal Circuit in *Oracle* recognized this possibility when remanding the case for retrial on the transformativeness issue. Under the *Connectix* holding on transformativeness, the jury could have reasonably found that Google's reimplementation of elements of the Java API was transformative.[120] Indeed, accepting Oracle's arguments to the contrary would largely preclude fair use as a viable defense in software reuse contexts, since the functional nature of software means that reusing software as software will inevitably implicate the very same computing functions for which the software was originally designed.[121]

B. Other Technology-Related Fair Use Rulings Have Affirmed Transformativeness Even When Whole Works Have Been Copied Without Alteration

Connectix is only one of numerous fair use precedents that have upheld technological transformativeness fair use defenses. These cases also support the argument that a defendant's reuse of an API or elements of an API can be transformative. In general, these cases found reuses of entire copyrighted works with little or no alteration were transformative and fair when the use involved an innovative technological purpose. Oracle has ignored these decisions and seeks to confine the meaning of "transformative" to a far narrower realm than these precedents support. Accepting such a narrow construction could eliminate the viability of the fair use defense in not only software cases, but technology cases more generally, thereby threatening the productive balance between copyright owners and follow-on innovators that the fair use defense helps achieve.

In *Field v. Google Inc.*, for example, Google's system for caching and displaying cache contents of websites was held to be transformative.[122] That use had a different purpose and character than the works

[119] *Id.* at 1522–24.

[120] Oracle Am., Inc. v. Google Inc., No. C 10-03561 WHA, 2016 WL 3181206, at *9 (N.D. Cal. June 8, 2016).

[121] *See generally* Clark D. Asay, *Transformative Use in Software*, 70 STAN. L. REV. ONLINE 9 (2017).

[122] 412 F. Supp. 2d 1106 (D. Nev. 2006).

originally had, in that Google's system enabled archiving, web page comparisons, and comprehension of search query results.[123]

In *Perfect 10, Inc. v. Amazon.com, Inc.*,[124] Google's use of copyrighted images to serve up thumbnail images in response to search queries was held to be transformative. The Ninth Circuit so held despite Google using scaled-down images in their entirety, in large part because Google's use achieved an innovative technological purpose with significant public benefit.[125]

Likewise, in *Authors Guild, Inc. v. HathiTrust*,[126] the Second Circuit upheld as highly transformative the copying of millions of copyrighted books in their entirety to create a searchable digital repository.[127] This use was transformative because it enabled new and innovative technological uses of the books, despite otherwise failing to change the contents of the books.[128] With the aid of a search engine, the HathiTrust digital library book corpus could be used for text and data mining purposes.[129] Researchers could formulate a search query to identify every book in that library that mentioned a particular concept, person, event, or place and find out on which pages of the books they would be able to find relevant materials.

Another text-mining fair use case that involved copying the whole of copyrighted works was *A.V. ex rel. Vanderhye v. iParadigms, LLC*.[130] In that case, the Fourth Circuit held that iParadigms' use of entire, unaltered copies of copyrighted works to detect plagiarism was transformative, in part because the use involved new technological purposes that provided significant public benefits.[131]

These decisions further support the view that reusing APIs can serve innovative technological purposes. Given these precedents, the Federal Circuit should conclude that the jury could reasonably find that Google's use of the Java API was transformative. For the Federal Circuit to hold that the reuse of an API cannot be transformative as a matter of law not only would be inconsistent with its prior ruling in the *Oracle* case, but also would shut the door to fair use defenses in future cases.

[123] *Id.* at 1118–19.

[124] 508 F.3d 1146 (9th Cir. 2007).

[125] *Id.* at 1165–66; *see also* Kelly v. Arriba Soft Corp., 336 F.3d 811, 818–19 (9th Cir. 2003) (coming to similar conclusions).

[126] 755 F.3d 87 (2d Cir. 2014).

[127] *Id.* at 97–98.

[128] *Id.*; *see also* Authors Guild v. Google Inc., 804 F.3d 202, 216–18 (2d Cir. 2015) (finding it transformative to digitize books to index their contents and serve up snippets of text in response to search queries).

[129] *HathiTrust*, 755 F.3d at 90–91.

[130] 562 F.3d 630 (4th Cir. 2009).

[131] *Id.* at 638–40.

C. Follow-On Innovators Are Justified in Using Significant Portions of Copyrighted Works if the Amount Taken Is Reasonable in Light of a Defendant's Transformative Purpose

Because prior case law makes clear that, as a matter of law, Google's use of the Java API could be transformative, the Federal Circuit should also be skeptical of Oracle's concerns that Google used too much of the Java API in its Android platform. In keeping with the Supreme Court's directives in *Campbell*, the transformative use inquiry should be closely linked with the third fair use factor's concern about the amount and substantiality of the taking. If the amount taken was reasonable in light of a transformative purpose, that use is more likely to be fair.[132]

So, if Google's use of the Java API declarations was transformative and the amount taken was reasonable in light of this transformative purpose, the jury could have reasonably found in favor of Google's fair use defense. Google used limited portions of the Java API as part of a highly innovative mobile device platform.[133] In building the Android platform, Google not only wrote its own implementing code for the portions of the Java API that it used, but it also created many new declarations to enable a vast array of additional, innovative smartphone functionalities.[134] Furthermore, the portions of the Java API that Google reimplemented may have helped preserve consistency of use within the larger Java developer community.[135] Given this, the jury could have reasonably concluded that Google's use of limited portions of the Java API was reasonable to achieve a transformative purpose.

It is important for future reusers of APIs that the jury's verdict of fair use in the *Oracle* case not be overturned on the crabbed view of fair use that Oracle has put forward. Instead, consistent with the Supreme Court's directive, reusers of APIs claiming a fair use defense should be able to use as much of an API as is reasonable in light of their transformative purposes.

[132] *See* Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 586–87 (1994) (recognizing "that the extent of permissible copying varies with the purpose and character of the use"); *HathiTrust*, 755 F.3d at 96 ("[W]e assess . . . whether the amount copied is reasonable in relation to the purported justifications for the use under the first factor.").

[133] Oracle Am., Inc. v. Google Inc., No. C 10-03561 WHA, 2016 WL 3181206, at *8–10 (N.D. Cal. June 8, 2016).

[134] *Id.* at *9.

[135] *Id.* at *11.

V. COMPUTER PROGRAM APIS ARE FUNCTIONAL CREATIONS THAT ENJOY BROADER FAIR USE RIGHTS THAN ARTISTIC OR OTHER HIGHLY EXPRESSIVE WORKS

The fair use provision of U.S. copyright law requires consideration of the nature of the work from which the defendant took some expression along with the purpose of the taking, the amount taken, and harm to the market for the work.[136] When defendants have raised fair use defenses in software copyright cases, the nature-of-the-work factor has been given considerable weight.

Google has argued that the Java API elements it used in the Android platform were highly functional and should, therefore, be subject to a broad scope of fair use.[137] By contrast, Oracle has contended that the Java API declarations and classes at issue are highly creative and that the nature-of-the-work factor must, consequently, disfavor Google's fair use defense as a matter of law.[138] Oracle's argument is a mistaken attempt to deflect attention away from the obvious fact that the work at issue in the *Oracle* case is software.

Under controlling Ninth Circuit precedents, functional writings, such as software, while protectable by copyright, enjoy a thinner scope of protection and a broader scope of fair use than highly expressive works, such as novels, music, dramas, and paintings.[139] The jury heard evidence that the declarations and classes of the Java API were functional.[140] By ruling in favor of Google's fair use defense, the jury seems to have decided that the declarations and classes were more functional than expressive. If the Federal Circuit overturns the jury verdict on fair use, in part on the ground that the Java API was highly creative and nonfunctional, that would conflict with Ninth Circuit precedents and would cast a dark shadow on fair use defenses in future software copyright cases.

A. Software Is Copyrightable, but Enjoys a Thin Scope of Protection Because of Its Functionality

The functionality of software as a limiting factor on copyright's scope has been well-recognized in Ninth Circuit case law. The leading case is *Apple Computer, Inc. v. Microsoft Corp.*,[141] in which the functionality of various features of a graphical user interface (GUI) nar-

[136] 17 U.S.C. § 107 (2012).

[137] Corrected Brief of Defendant-Appellee/Cross-Appellant Google Inc. at 49–50, Oracle Am., Inc. v. Google Inc., (Nos. 117-1118, 1202) (Fed. Cir. May 22, 2017).

[138] Oracle Brief, *supra* note 11, at 39–43.

[139] *See, e.g.*, Apple v. Microsoft, 35 F.3d 1435, 1444–45 (9th Cir. 1994).

[140] *Oracle*, 2016 WL 3181206, at *10.

[141] 35 F.3d 1435 (9th Cir. 1994).

rowed the scope of copyright in Apple's operating system (OS).[142] The Ninth Circuit agreed with the district court that because of this functionality, the GUI components of the Apple OS were "entitled only to limited protection and should be compared for virtual identity follow[ing] from its analytic dissection" of the specific elements for which Apple was seeking protection.[143] The Ninth Circuit approvingly reviewed the district court's application of several limiting principles of copyright, including functionality, standardization, *scenes a faire*, and merger, to the Apple GUI on a feature-by-feature basis.[144]

The Ninth Circuit rejected Apple's contention that its GUI was highly creative and entitled to broad protection for its "look and feel."[145] The court noted that "unlike purely artistic works such as novels and plays, [GUIs] generated by computer programs are partly artistic and partly functional."[146] To the extent that the GUI features were functional or constrained by external factors, those elements must remain outside of copyright's boundaries.[147] The court concluded that the district court had properly taken into account "the functional aspects of [GUIs] and the analogous range of protection available for compilations."[148]

The Ninth Circuit in *Apple* cited approvingly to *Altai*, the Second Circuit's major software copyright decision.[149] Like the Ninth Circuit, the *Altai* court recognized that "the essentially utilitarian nature of a computer program" complicates the task of distinguishing its protectable from unprotectable elements.[150] It directed courts to filter out unprotectable functional elements, as well as ideas and standard techniques, before proceeding to the comparison stage of the infringement analysis.[151] The court recognized that this would narrow the scope of copyright protection, but "that result flows from applying, in accordance with Congressional intent, long-standing principles of copyright law."[152]

Like the Ninth Circuit, the Second Circuit regards computer programs as very different in nature from aesthetic works. The functional character of programs means that they "hover" near "the elusive

[142] *Id.* at 1443–46.

[143] *Id.* at 1446; *see also id.* at 1438–39.

[144] *Id.* at 1444–47.

[145] *Id.* at 1439.

[146] *Id.* at 1444.

[147] *Id.* at 1443–46.

[148] *Id.* at 1442 n.10.

[149] *Id.* at 1445 (citing Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 706–11 (2d Cir. 1992)).

[150] *Altai*, 982 F.2d at 704.

[151] *Id.* at 706–08.

[152] *Id.* at 712.

boundary line described in § 102(b)."[153] Owing to the hybrid nature of software — as both a "literary expression" and a "highly functional, utilitarian component in the larger process of computing" — copyright provides only a "weak barrier" of protection for programmers.[154] Although CAI and its amici urged the court to construe the scope of software copyrights broadly, the Second Circuit thought this would have "a corrosive effect on certain fundamental tenets of copyright doctrine."[155] To get exclusive rights in the more functional elements of software, the Second Circuit thought it more appropriate for developers to seek patents.[156]

B. The Nature-of-the-Work Factor Generally Favors Fair Use in Software Cases

The functionality of software and of API procedures weighed in favor of the fair use defenses in the *Connectix* and *Accolade* cases.[157] The Federal Circuit's previous *Oracle* decision acknowledged that Ninth Circuit decisions had taken the functionality of software into account in those fair use cases.[158]

In *Connectix*, the Ninth Circuit began its fair use analysis with consideration of the nature-of-the-work factor, signaling therewith that this factor carries greater weight in software fair use cases because of software's functional nature. In contrast, courts generally give little attention to the nature-of-the-work factor in run-of-the-mill fair use analyses.[159] But the *Connectix* court emphasized this factor, observing that "Sony's BIOS lies at a distance from the core [of copyright] because it contains unprotected aspects that cannot be examined without copying."[160] The court quoted approvingly to its earlier *Accolade* decision that accorded software a "lower degree of protection than more traditional literary works."[161] Fair use was an appropriate way to "preserve[] public access to the ideas and functional elements

[153] *Id.* at 704.

[154] *Id.* at 712.

[155] *Id.*

[156] *Id.* That software API designs can be functional is evident from the fact that some have been patented. *See* Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974, 996 (N.D. Cal. 2012).

[157] Sony Computer Entm't v. Connectix Corp., 203 F.3d 596, 605 (9th Cir. 2000); Sega Enters. v. Accolade, 977 F.2d 1510, 1526 (9th Cir. 1992). The word "functional" appears almost 60 times in the *Accolade* decision, and in each context, the term is viewed as a limit on the scope of copyright (as well as trademark) protection.

[158] Oracle Am., Inc. v. Google Inc., 750 F.3d at 1375–76 (Fed. Cir. 2014).

[159] *See* Robert Kasunic, *Is That All There Is? Reflections on the Nature of the Second Fair Use Factor*, 31 COLUM. J.L. & ARTS 529 (2008).

[160] *Connectix*, 203 F.3d at 603.

[161] *Id.* (quoting *Accolade*, 977 F.2d at 1526).

embodied in copyrighted computer software."[162] The court regarded the nature-of-the-work factor to "strongly favor[]" Connectix's fair use defense.[163]

In *Accolade*, as in *Connectix*, the Ninth Circuit regarded the nature-of-the-work factor to be "important to the resolution of cases such as the one before us," especially given the "ultimate use to which [the defendant] put the functional information" it derived from the plaintiff's programs in developing its own program.[164] The Ninth Circuit observed:

> The second statutory [fair use] factor . . . reflects the fact that not all copyrighted works are entitled to the same level of protection . . . Works of fiction receive greater protection than works that have strong factual elements, such as historical or biographical works, or works that have strong functional elements, such as accounting textbooks.[165]

Copyright protection "does not extend to the ideas underlying a work or to the functional or factual aspects of the work," which is why "[u]nder the Copyright Act, if a work is largely functional, it receives only weak protection."[166] This result, said the court, was "neither unfair nor unfortunate. It is the means by which copyright advances the progress of science and art."[167]

The functional nature of software has favored fair use in the past because of the desirability of enabling second-comers to build on the functional elements of existing programs in creating new works of authorship. The jury's verdict strongly suggests that it regarded the nature-of-the-work factor as weighing in favor of fair use. The Federal Circuit should not disturb that finding. Courts in future software copyright fair use cases should also recognize the highly functional nature of software as a significant consideration in assessing and weighing the importance of the nature-of-the-work factor as part of the fair use inquiry.

[162] *Connectix*, 203 F.3d at 603.

[163] *Id.* at 605.

[164] *Accolade*, 977 F.2d at 1522.

[165] *Id.* at 1524 (citations omitted).

[166] *Id.* at 1524, 1527.

[167] *Id.* (quoting Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 350 (1991)); *see also* Bikram's Yoga College of India, L.P. v. Evolation Yoga, LLC, 803 F.3d 1032, 1037–41 (9th Cir. 2015) (finding yoga sequence unprotectable as a functional work under § 102(b) regardless of its aesthetic considerations).

VI. Conclusion

The outcome of the *Oracle v. Google* case now before the Federal Circuit will have a significant impact on fair use law, software copyright law, and the balances struck between copyright owners' legitimate interests in protecting their rights and the interests of second-comers in being able to build on earlier innovations that intellectual property laws aspire to achieve. In reliance on pervasive industry norms and decisions such as *Accolade* and *Altai,* software developers have for more than two decades been confident that they could reimplement APIs in independently created programs without infringing copyright laws. During those decades, competition and innovation in the software industry have thrived, and consumers have greatly benefited from the wide array of interoperable software products and services.

If the Federal Circuit decides to embrace Oracle's arguments regarding the role of subjective mental states, transformativeness, and fair use's nature-of-the-work factor, that would severely constrain, if not entirely defeat, the ability of follow-on innovators to raise fair use defenses in future software copyright API cases. The Federal Circuit should affirm the lower court's denial of Oracle's motion for a judgment as a matter of law on these issues.

In order to promote ongoing innovation and competition in the software industry, courts should preserve a role for fair use in regulating the reuse of program APIs. Oracle may be greatly disappointed that it was unable to capitalize on using Java technologies in smartphones, but copyright law should not be used to thwart progress and innovation in software by others.