

**ORACLE V. GOOGLE: SETTING A STANDARD OR HANDICAPPING AN
INDUSTRY?**

Tyler J. Demasky*

For years the world of software programming has been operating under the assumption that both the declaration code and the structure, sequence, and organization (SSO) of Java application program interfaces (APIs) were not copyrightable and were therefore free for all to use. However, when Oracle sued Google in 2014 for Google’s use of Java APIs in the Android Operating System (OS), the Federal Circuit held that the declaration code and the SSO of APIs are protected by copyright, and remanded the case back to the district court. The jury found in favor of Google—who had to rely on a “fair use” defense—and Oracle again appealed to the Federal Circuit. The Federal Circuit should affirm the jury’s verdict because the district court’s interpretation of the statutory factors align more closely with the nature of software programming. Further, reversing the jury’s verdict could potentially set a negative, restrictive precedent and have detrimental effects on the software industry.

* J.D. Candidate, University of North Carolina School of Law, 2018. The author would like to thank all of the JOLT V.18 editors, but especially his Notes Editor, Nick Hanna, and his Editor-In-Chief, Shannon O’Neil. He would also like to thank Jeffrey Kaufman and Scott Peterson of Red Hat, Inc. for assisting the author in his understanding of the software technology presented in this article.

I. INTRODUCTION	2
II. HISTORICAL “FAIR USE”	4
<i>A. The Supreme Court’s “Fair Use” Interpretations.....</i>	<i>5</i>
<i>B. The Ninth Circuit’s “Fair Use” Interpretations.....</i>	<i>8</i>
III. JAVA	10
<i>A. What is Java?.....</i>	<i>10</i>
<i>B. The Structure of Java Code</i>	<i>11</i>
<i>C. Copyright Law and Java.....</i>	<i>12</i>
1. <i>Literal Elements.....</i>	<i>13</i>
2. <i>Non-literal Elements</i>	<i>13</i>
IV. ORACLE V. GOOGLE: HISTORY AND HOLDINGS	14
<i>A. Leading up to the Initial Oracle v. Google Case.....</i>	<i>15</i>
<i>B. District Court and Federal Circuit Results of the Initial Oracle v. Google Case.....</i>	<i>16</i>
<i>C. District Court Results from the Remanded Oracle v. Google Case.....</i>	<i>17</i>
V. RULING IMPLICATIONS.....	18
<i>A. What Affirming the Current Ruling Could Mean.....</i>	<i>18</i>
1. <i>The Statutory Factors as Charged to the Jury.....</i>	<i>19</i>
2. <i>Why the Charged Interpretations Should Be Affirmed</i>	<i>21</i>
<i>B. What Reversing the Verdict Could Mean</i>	<i>25</i>
1. <i>Effect on Copyright Law.....</i>	<i>25</i>
2. <i>Effect on the Software Industry.....</i>	<i>28</i>
VI. CONCLUSION	32

I. INTRODUCTION

The law often has to play catch-up with modern changes in society. This catch-up is perhaps most prominent in intellectual property and technology. However, the Federal Circuit¹ has a chance to bridge the gap. In 2010, Oracle sued Google for copyright infringement of Java “application program interfaces” (“APIs”) owned by Oracle in Google’s Android operating system

¹ See *infra* Part IV. Section B. explaining the Federal Circuit’s jurisdiction.

(“OS”).² The Federal Circuit remanded the case to the district court to be retried on the basis that APIs are copyrightable,³ where Google successfully defended its use of APIs under the “fair use” doctrine.⁴

Oracle has again appealed to the Federal Circuit,⁵ and the court will consider a potentially significant and impactful decision. Upholding the jury verdict could potentially set a new, more liberal test for “fair use” when dealing with software programming,⁶ which in turn could help lessen the gap between law and technology. Reversing the decision could have the opposite effect, resulting in more stringent copyright laws and debilitating innovation for an entire industry.

This Recent Development examines the implications of a Federal Circuit ruling by proceeding in six parts. Part II provides an overview of the “fair use” doctrine and its historical interpretations. Part III introduces and explains Java, Java APIs, and how copyright law attaches to the literal and non-literal elements of APIs. Part IV addresses the history of the case, both inside and outside of court. Part V first analyzes the current verdict and addresses why the district court’s verdict should be affirmed. It then assesses what reversing the verdict could mean legally, and how that might affect the software industry in particular. Finally, Part VI concludes by reiterating the possible consequences of Oracle’s appeal to the Federal Circuit on the cloudy area of software copyright law.

² See *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 975 (N.D. Cal. 2012).

³ See *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1381 (Fed. Cir. 2014).

⁴ See Final Judgment, *Oracle Am., Inc. v. Google Inc.*, No. C-10-03561 WHA (N.D. Cal. June 8, 2016).

⁵ See Notice of Appeal to the Fed. Cir. by Oracle Am., Inc., docket number 3:10-cv-03561, entry # 2071, filed on 10/26/2016.

⁶ See *infra* Part V. Section A.1.

II. HISTORICAL “FAIR USE”

According to the Copyright Act of 1976, determining whether use of a work is “fair use” involves the consideration of several factors, including the following:

- (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes (“purpose”);
- (2) the nature of the copyrighted work (“nature”);
- (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole (“amount”); and
- (4) the effect of the use upon the potential market for or value of the copyrighted work (“market effect”).⁷

While the Copyright Act lays out these four factors, it does not define them.⁸ Defining the factors, or even contributing additional factors, is left to the courts.⁹ It is “an equitable rule of reason,”¹⁰ meaning that results may vary, causing some to refer to the “fair use” defense as “the most troublesome [doctrine] in the whole law of copyright.”¹¹ Different courts may weigh factors differently, and some courts may even weigh factors differently from case to case.¹² This section discusses “fair use” by focusing on: (A) the Supreme Court’s interpretations of fair use in *Sony Corporation of*

⁷ 17 U.S.C. § 107 (2012).

⁸ *Id.*

⁹ See Final Charge to the Jury (Phase One) & Special Verdict Form, at 12, Oracle Am., Inc. v. Google Inc., No. C-10-03561 WHA (N.D. Cal. June 8, 2016) (“[T]he statute includes several examples of some types of uses that may be found to be fair uses, but that list is not exhaustive or exclusive.”) [hereinafter Final Charge].

¹⁰ Harper & Row, Publs. v. Nation Enters., 471 U.S. 539, 560 (1985) (quoting H.R. Rep. No. 94-1476, at 66 (1976)). An equitable rule of reason means that the doctrine is fact based, and an accurate application of the doctrine requires a case-by-case analysis. See *id.*

¹¹ Dellar v. Samuel Goldwyn, Inc., 104 F.2d 661, 662 (2d Cir. 1939).

¹² See Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 448-51, 454-55 (1984); see also Harper & Row, 471 U.S. at 560-68; Sega Enters. v. Accolade, Inc., 977 F.2d 1510, 1521-27 (9th Cir. 1992).

*America v. Universal City Studios, Inc.*¹³ and *Harper & Row, Publishers v. Nation Enterprises*,¹⁴ and (B) the Court of Appeals for the Ninth Circuit's ("Ninth Circuit") application of Supreme Court guidance in *Sega Enterprises v. Accolade, Inc.*¹⁵

A. *The Supreme Court's "Fair Use" Interpretations*

In *Sony Corporation of America*,¹⁶ the Supreme Court held that Sony's Betamax¹⁷ tapings of previously broadcast television programming constituted fair use, putting the emphasis most heavily on the first statutory factor (purpose), followed respectively by the fourth (market effect), and third (amount).¹⁸ Sony produced and sold Betamax, a home video tape recorder, which allowed customers to record television programs.¹⁹ Universal Studios sued Sony for copyright infringement based on Sony's customers using Betamax to record Universal Studios' televised copyrighted broadcasts, and Sony raised a "fair use" defense.²⁰ The district court held in favor of Sony, but the appellate court reversed.²¹ The U.S. Supreme Court granted certiorari, and reversed the appellate court's decision, holding for Sony.²²

The Supreme Court appeared to put the most emphasis on the first factor (purpose).²³ Most notably, the Court addressed the fact that the primary use for the Betamax tapes was private home use, which "must be characterized as a noncommercial, nonprofit activity."²⁴ Because the purpose was for private, personal viewing in one's home at a later time, the Court felt there was a presumption

¹³ See *Sony*, 464 U.S. at 448-51, 454-55.

¹⁴ See *Harper & Row*, 471 U.S. at 560-68.

¹⁵ See *Sega*, 977 F.2d at 1521-27.

¹⁶ *Sony*, 464 U.S. at 448-51, 454-55.

¹⁷ Betamax was a home video tape recorder produced by Sony. *Id.* at 422.

¹⁸ See *Sony*, 464 U.S. at 448-51, 454-55.

¹⁹ *Id.* at 419.

²⁰ *Id.* at 420, 424.

²¹ *Id.* at 420.

²² *Id.* at 421.

²³ *Sony*, 464 U.S. at 449.

²⁴ *Id.*

of fairness in the use.²⁵ Additionally, the Court's decision was based on the fact that because the nature of the public broadcasts invited those customers to watch the programs in their entirety free of charge, it had no demonstrable effect on Universal's potential market.²⁶

Just one year later, in *Harper & Row*,²⁷ the Supreme Court held that The Nation's ("Nation") verbatim copying of excerpts from former President Ford's unpublished memoirs did not constitute fair use.²⁸ Instead, the Court placed the weight of its decision on the fourth statutory factor (market effect), followed respectively by the third (amount) and second (nature).²⁹ Nation was a magazine that had originally received the rights from Ford to license prepublication excerpts from his memoirs and in turn negotiated with Time Magazine for the opportunity for Time to publish those excerpts.³⁰

After negotiations, Nation obtained a leaked copy of the memoir and published a piece containing an excerpt of some 300-400 words comprising verbatim quotes from the manuscript.³¹ Nation released the article just before the Time article was supposed to print, "scooping"³² the Time article.³³ As a result, Time refused to pay the remaining balance negotiated for the excerpts.³⁴ Nation sued to recover the balance, but the district court held for Time, stating that a copyright protected the Ford memoirs, and Nation infringed upon that copyright.³⁵ The court of appeals

²⁵ *Id.*

²⁶ *Id.* at 450.

²⁷ See *Harper & Row, Publr. v. Nation Enters.*, 471 U.S. 539, 560-68 (1985).

²⁸ See *id.*

²⁹ See *id.*

³⁰ *Id.* at 542-53.

³¹ *Id.* at 543, 548.

³² *Id.* at 542 (implying that "scooping" means to take the newsworthiness of the information away from the Time article, rendering it less impactful and thus potentially reducing marketability of the article).

³³ *Harper & Row*, 471 U.S. at 543.

³⁴ See *id.* at 543.

³⁵ *Id.* at 543-44.

reversed, stating that Nation's publication was "fair use," but the U.S. Supreme Court reversed that holding.³⁶

Contrary to the *Sony* case, the Court in *Harper & Row*, held "[the fourth] factor is undoubtedly the single most important element of 'fair use,'"³⁷ because, to be applied properly, fair use must be "limited to copying by others which does not materially impair the marketability of the work which is copied."³⁸ Like in the *Sony* case, a showing of adverse effect on the potential market could negate the "fair use."³⁹ The Court reasoned that the relevant market in this case was to be the first to release new information on Ford, which is what Time intended to do.⁴⁰ Nation's publishing of the excerpts before Time, which could be recognized as Ford speaking and not Nation,⁴¹ "directly competed for a share of the market for prepublication excerpts."⁴²

For the third and second factors, ordered according to the weight given to them by the Court, the focus was on the qualitative amount of copyrighted work used, rather than the quantitative amount,⁴³ and the nature of that work.⁴⁴ Nation verbatim published a small quantitative amount of the manuscript, but captured "what was essentially the heart" of the work.⁴⁵ An editor of Time stated that the chapters quoted were "the most interesting and moving parts of the entire manuscript" and were "the most powerful passages in those chapters."⁴⁶ The Court noted that "[t]he fact that

³⁶ *Id.* at 542, 544.

³⁷ *Id.* at 566.

³⁸ *Id.* at 566-567 (quoting Melvill B. Nimmer and David Nimmer, Nimmer on Copyright, § 1.10[D] at 1-87 (Rev. Ed. 2011)).

³⁹ *See Harper & Row*, 471 U.S. at 567.

⁴⁰ *Id.* at 603.

⁴¹ *Id.* at 568.

⁴² *Id.*

⁴³ *Id.* at 565.

⁴⁴ *See id.* at 563.

⁴⁵ *Harper & Row*, 471 U.S. at 564-65 (quoting *Harper & Row, Publs. v. Nation Enters.*, 557 F. Supp. 1067, 1072 (S.D.N.Y. 1983)).

⁴⁶ *Harper & Row*, 471 U.S. at 565.

a work is unpublished is a critical element of its ‘nature,’” and that the scope of “fair use” should be narrowed in such cases.⁴⁷

The Supreme Court’s position on the importance of each of the factors appears ambiguous, offering little guidance to lower courts. The Ninth Circuit confirmed this ambiguity in *Sega v. Accolade*,⁴⁸ choosing to emphasize different statutory factors.

B. The Ninth Circuit’s Fair Use Interpretations

In *Sega*, the Ninth Circuit held that Accolade’s breaking down of Sega programming was fair use.⁴⁹ The Ninth Circuit followed the Supreme Court in *Sony*, putting the most weight on the first statutory factor (purpose), followed by the fourth factor (market effect), but went its own way by focusing additionally on the second factor (nature).⁵⁰ Accolade was attempting to manufacture video games for various game consoles, but wanted compatibility with the Sega Genesis system.⁵¹ To determine the compatibility requirements, Accolade disassembled Sega video game software to find the code needed for compatibility.⁵² It then copied the code necessary for compatibility, but wrote its own procedures and developed its own games.⁵³ Sega sued for copyright infringement of its code.⁵⁴ The district court found for Sega, but the Ninth Circuit reversed, saying Accolade’s use of the code was “fair use.”⁵⁵

The Ninth Circuit put a lot of weight on whether the use was for commercial purposes.⁵⁶ However, instead of looking at the ultimate result of Accolade developing games to sell that were compatible with the Genesis system, the court focused on

⁴⁷ *Id.* at 564.

⁴⁸ *See Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510, 1521-27 (9th Cir. 1992).

⁴⁹ *See id.* at 1527.

⁵⁰ *See id.* at 1521-27.

⁵¹ *Id.* at 1514.

⁵² *Id.* at 1514-15.

⁵³ *Id.* at 1515.

⁵⁴ *Sega*, 977 F.2d at 1516.

⁵⁵ *See id.* at 1517, 1527-28.

⁵⁶ *See id.* at 1522.

Accolade's narrowed purpose of seeking to understand the functional requirements of the Genesis system.⁵⁷ Because this was an educational purpose, the court concluded that Accolade's use was "for a legitimate, essentially non-exploitative purpose" and could thus be described as being of "minimal significance" commercially.⁵⁸ The court then tied the purpose and character to the fourth factor: the effect on the potential market.⁵⁹

No significant effect on the market was found in *Sega* because, unlike in *Harper & Row*, Accolade was not attempting to "scoop" Sega's release of particular video games.⁶⁰ Rather, Accolade wanted to break into the market for Genesis-compatible video games.⁶¹ The nature of video games is such that the purchasing of one game of a specific type does not preclude the same consumer from also purchasing a different game of the same or similar type.⁶² The court used a test⁶³ from the Second Circuit to boil the nature of the code down to the core ideas of the program to determine that the copied code was mostly functional elements, weighing in favor of fair use.⁶⁴

The weight and focus of each statutory factor will differ depending on the specifics of the case at bar, but it appears to be possible for a court to place greater weight on one factor over others. An appeal to the Federal Circuit could introduce the possibility of factor rankings.⁶⁵ However, in order to address the implications of a Federal Circuit decision on "fair use" factors, the next section first necessarily explains the technologies at issue in *Oracle v. Google: Java and Java APIs*.

⁵⁷ *Id.*

⁵⁸ *Id.* at 1522-23.

⁵⁹ *Id.* at 1523.

⁶⁰ *See Sega*, 977 F.2d at 1523.

⁶¹ *Id.*

⁶² *Id.*

⁶³ *See infra* Part III. Section C.2.

⁶⁴ *See Sega*, 977 F.2d at 1524-26.

⁶⁵ *See infra* Part V. Section A.

III. JAVA

This section provides a background on Java by (A) explaining the substance of Java, (B) describing the structure of Java, and (C) demonstrating how copyright law applies to the structure and organization of Java.

A. *What is Java?*

Java is a programming language initially developed and owned by Sun Microsystems, Inc. (“Sun”).⁶⁶ Like all programming languages, Java has a combination of words, symbols, and other units arranged according to specified syntax rules to create various instructions.⁶⁷ Generally, this is referred to as source code, the version of a program that can be read by humans.⁶⁸ However, Java differs from other programming languages. Instead of the computer directly interpreting Java source code into something the computer can understand—binary machine code⁶⁹—Java code is first compiled into bytecode⁷⁰ and then interpreted by a Java Virtual Machine (“JVM”).⁷¹ The JVM is an executable program on the computer that interprets the bytecode into machine code, executing specific operations.⁷² This intermediate step allows the Java code to be hardware/platform independent.⁷³ The source code

⁶⁶ Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1347 (Fed. Cir. 2014).

⁶⁷ See *id.* at 1348; see generally Monica Pawlan, *Essentials of the Java Programming Language, Part I*, ORACLE (March 1999), <http://www.oracle.com/technetwork/java/index-138747.html> (providing an introduction to the Java programming language).

⁶⁸ See *Oracle*, 750 F.3d at 1348.

⁶⁹ Binary machine code is a collection of 0s and 1s that form instructions a computing device can understand. See *Oracle*, 750 F.3d at 1347.

⁷⁰ Bytecode is intermediate, machine-independent, object code that is the result of a program compiling the inputted Java source code. See Java Bytecode, TECHOPEDIA, <https://www.techopedia.com/definition/7866/java-bytecode> (last visited Nov. 17, 2016); see also Bill Venner, *Bytecode Basics: A First Look at the Bytecodes of the JavaVirtual Machine*, JAVAWORLD, (Sep. 1, 1996, 1:00 AM), <http://www.javaworld.com/article/2077233/core-java/bytecode-basics.html>.

⁷¹ See *Oracle*, 750 F.3d at 1348.

⁷² See *id.*

⁷³ See *id.* at 1347.

and the machine code are both generally “literal elements”⁷⁴ of a computer program.⁷⁵

B. The Structure of Java Code

Programs work by calling specific operations to do different tasks, and the codes for those specific operations are called “Methods.”⁷⁶ Those Methods, along with variables and other elements on which the methods operate, make up different “Classes.”⁷⁷ At the broadest level, those Classes are organized into “Packages.”⁷⁸ There are two components to Packages: the declaring code and the implementing code.⁷⁹ The APIs at issue in the *Oracle v. Google* case are Packages, the high-level, general pieces of code.⁸⁰ The district court analogized the structure as follows: “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.”⁸¹ To continue the analogy, the declaring code would be like the title of that how-to chapter, and the implementing code would be the text of the chapter. This is the APIs’ SSO, and it is considered a “non-literal element”⁸² of a computer program.⁸³

APIs allow Java programmers to use prewritten programs to build functions into their own Java applications, rather than having to start over from scratch.⁸⁴ As stated above, APIs consist of

⁷⁴ See *infra* Part III. Section C.1.

⁷⁵ See *Oracle*, 750 F.3d at 1355.

⁷⁶ See *id.* at 1349; see also Pawlan, *supra* note 67 (describing Methods under *Lesson 2: Building Applications*).

⁷⁷ See *Oracle*, 750 F.3d at 1349; see also Pawlan, *supra* note 67 (describing Classes under *Lesson 2: Building Applications*).

⁷⁸ See *Oracle*, 750 F.3d at 1349; see also Pawlan, *supra* note 67 (describing Packages under *Lesson 2: Building Applications*).

⁷⁹ See *Oracle*, 750 F.3d at 1349.

⁸⁰ See Google’s Trial Brief at 1, *Oracle Am., Inc. v. Google Inc.*, No. C-10-03561 WHA (N.D. Cal. June 8, 2016).

⁸¹ *Oracle*, 750 F.3d at 1349.

⁸² See *infra* Part III. Section C.2.

⁸³ See *Oracle*, 750 F.3d at 1355-56.

⁸⁴ *Id.* at 1349.

declaring code and implementing code.⁸⁵ The declaring code identifies the method body and specifies inputs, names, and other functionalities. For example, using the declaration “public static int max(int x, int y)” signals the objective of returning the maximum of x and y indicated between the parentheses.⁸⁶ The code that actually calculates the maximum value and returns it is called the implementing code.⁸⁷ The parties in *Oracle v. Google* are specifically arguing about whether Google’s use of the declaring code and SSO of the Java APIs was fair.⁸⁸

C. Copyright Law and Java

This section addresses how copyright law applies to (1) literal elements, and (2) non-literal elements of computer programs. Under the Copyright Act of 1976, copyright protection exists in “original works of authorship fixed in any tangible medium of expression,” including literary works.⁸⁹ Copyright protection does not, however, exist for “any idea, procedure, process, system, method of operation, concept, principle, or discovery”⁹⁰ This means the law protects the expression itself, rather than the underlying idea.⁹¹ So for a particular program component to be protected, it must qualify as an expression of an idea, not the idea itself.⁹² If the expression of an idea were inseparable from the idea

⁸⁵ *Id.*

⁸⁶ *See id.* (indicating that “public” shows a general accessibility of the method, “static” means the method can be called at any time, and “int” signifies the method will return an integer).

⁸⁷ *See id.* at 1350.

⁸⁸ *See* Google’s Trial Brief, *supra* note 80, at 8.

⁸⁹ 17 U.S.C. § 102(a)(1) (2012).

⁹⁰ 17 U.S.C. § 102(b).

⁹¹ *See Oracle*, 750 F.3d at 1354.

⁹² *See Johnson Controls, Inc. v. Phoenix Control Sys.* 886 F.2d 1173, 1175 (9th Cir. 1989). For example, a “jeweled bee pin” would be an idea, and actually creating a specific jeweled bee pin would be an expression of that idea. *See Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971). The line between an idea and an expression is not always easy to draw. *See id.* (“The critical distinction between ‘idea’ and ‘expression’ is difficult to draw. As Judge Hand candidly wrote, ‘Obviously, no principle can be stated as to when an imitator has gone beyond copying the “idea,” and has borrowed its

itself—perhaps due to a limited number of ways to properly express the idea—then the expression would not benefit from copyright protection.⁹³

1. *Literal Elements*

Although it is generally understood that copyright protection does not apply to names, titles, or short phrases,⁹⁴ courts have consistently held that copyright protection can apply to literal elements of a computer program.⁹⁵ In many cases, there are numerous ways to write programs and different names for a declaration.⁹⁶ The various options demonstrate creativity in choosing which declaration or specific method for executing a program, qualifying those literal elements as expressions and permitting them copyright protection.⁹⁷

2. *Non-literal Elements*

Partly for reasons similar to the literal elements above, courts have construed SSOs as expressions and granted them copyright protection.⁹⁸ Additionally, the Ninth Circuit endorsed a three-step

“expression.””) (quoting *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960)).

⁹³ See *Johnson Controls*, 886 F.2d at 1175; see also *Apple Computer, Inc. v. Formula Int’l, Inc.*, 725 F.2d 521, 525 (9th Cir. 1984).

⁹⁴ See UNITED STATES COPYRIGHT OFFICE, Circular 34, COPYRIGHT PROTECTION NOT AVAILABLE FOR NAMES, TITLES, or SHORT PHRASES (Oct. 2015).

⁹⁵ See *Oracle*, 750 F.3d at 1355; see also *Computer Assocs. Int’l v. Altai*, 982 F.2d 693, 702 (2d Cir. 1992); *Johnson Controls*, 886 F.2d at 1175; *Apple*, 725 F.2d at 525.

⁹⁶ See *Apple*, 725 F.2d at 525. When deciding copyrightability of declaration codes, the Federal Circuit stated: “The [district 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 *Oracle*, 750 F.3d at 1362. The analysis here presented by the court is too formalistic for the field of computer sciences, as it fails to consider the commonplace of certain declaration codes within the field. But, copyrightability of APIs is for another court at another time.

⁹⁷ See *Oracle*, 750 F.3d at 1356; see also *Apple*, 725 F.2d at 525.

⁹⁸ See *Oracle*, 750 F.3d at 1355; see also *Johnson Controls*, 886 F.2d at 1175.

test developed by the Second Circuit and adopted by several circuits as another way to determine if non-literal elements are expressions.⁹⁹ The test is called an “abstraction-filtration-comparison” test (“AFC test”), and it rejects both the “notion that anything that performs a function is necessarily uncopyrightable” and the notion 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.”¹⁰⁰ The test is as follows:

In the abstraction step, the court “first breaks down the allegedly infringed program into its constituent structural parts.”¹⁰¹ In the filtration step, the court “sifts out all non-protectable material,” including ideas and “expression that is necessarily incidental to those ideas.”¹⁰² In the final step, the court compares the remaining creative expression with the allegedly infringing program.¹⁰³

In one swoop, this test addresses both a copyrightability analysis and an infringement analysis.¹⁰⁴ The next section describes the role of literal and non-literal elements in the copyright infringement case brought by Oracle against Google for Google’s use of Java APIs.¹⁰⁵

IV. *ORACLE V. GOOGLE*: HISTORY AND HOLDINGS

This part introduces the history of the *Oracle v. Google* case by discussing (A) what led to the initial case, (B) the district court and Federal Circuit’s rulings in the initial case, and (C) the district court’s ruling upon remand.

⁹⁹ See *Oracle*, 750 F.3d at 1357.

¹⁰⁰ *Id.*

¹⁰¹ *Id.*

¹⁰² *Id.*

¹⁰³ *Id.* (citations omitted) (citing *Computer Assocs. Int’l v. Altai*, 982 F.2d 693, 706 (2d Cir. 1992)).

¹⁰⁴ *Id.* at 1357.

¹⁰⁵ See *supra* Part III. Section B. for an explanation of Java APIs.

A. Leading up to the Initial Oracle v. Google Case

In 2005, Google acquired Android Industries with the intent of developing the Android mobile OS.¹⁰⁶ Google initially tried to license the use of Java from Sun, but negotiations eventually fell through.¹⁰⁷ Google continued to develop Android with the assistance of Open Handset Alliance (“OHA”), using freely available Java language and other open source materials.¹⁰⁸ Google created a three-layered system, which runs the Android OS, utilizing open source code from existing sources, including Java language API libraries available from the Apache Software Foundation.¹⁰⁹

The bottom layer of Google’s system—the core operating system—utilizes the open source Linux kernel.¹¹⁰ For the middle layer, Google developed its own virtual machine—“Dalvik”—to execute programs in Java and other languages.¹¹¹ The upper layer consisted of API packages.¹¹² Google wrote over one hundred of its own API packages, but also used thirty-seven Java APIs relevant to a smartphone platform for the Android OS.¹¹³ Google copied the declaring code for those APIs verbatim, in order to allow application developers to find those functionalities by the same names used in Java within the Android OS.¹¹⁴

While Google admitted to copying the declaring code for the APIs and the SSO of the packages reflected in the declaring code,¹¹⁵ it “re-implemented” the APIs by writing and utilizing its

¹⁰⁶ See Google’s Trial Brief, *supra* note 80, at 2.

¹⁰⁷ See *id.* at 3-4.

¹⁰⁸ See *id.* at 4.

¹⁰⁹ See *id.* at 4-5.

¹¹⁰ See *id.* at 5. The Linux kernel is the core of the Linux operating system, an open source operating system developed by Linus Torvald. See *Kernel Definition*, THE LINUX INFORMATION PROJECT, <http://www.linfo.org/kernel.html> (last visited Oct. 25, 2016); see also *Linux Definition*, THE LINUX INFORMATION PROJECT, <http://www.linfo.org/linuxdef.html> (last visited Oct. 25, 2016).

¹¹¹ See Google’s Trial Brief, *supra* note 80, at 5.

¹¹² See *id.*

¹¹³ See Google’s Trial Brief, *supra* note 80, at 5.

¹¹⁴ See *Oracle*, 750 F.3d at 1339, 1350-51.

¹¹⁵ See Google’s Trial Brief, *supra* note 80, at 8.

own implementing code.¹¹⁶ In 2010, Oracle purchased Sun, and almost immediately sued Google for copyright infringement based on Android's use of the thirty-seven Java APIs.¹¹⁷ Oracle was asserting a patent infringement claim and copyright infringement for the method declarations (literal elements) and SSO (non-literal elements) of those API packages.¹¹⁸

B. District Court and Federal Circuit Results of the Initial Oracle v. Google Case

The district court originally found no patent infringement and held that APIs were not subject to copyright protection,¹¹⁹ but the jury hung on Google's fair use defense.¹²⁰ The district court concluded that the declaring code was uncopyrightable because it employed short phrases, the idea and expression merged, and the SSO was uncopyrightable as a "method of operation" under § 102(b) of the Copyright Act.¹²¹ Because the jury was unable to reach a conclusion on the fair use defense, the court never performed any fact finding on the defense for final judgment.¹²² Oracle appealed the district court's decision to the Federal Circuit.¹²³ The decision was appealed to the Federal Circuit because of the patent infringement claim, although Ninth Circuit law governed the copyright claim.¹²⁴

After de novo review,¹²⁵ the Federal Circuit reversed the lower court's decision, holding that the declaring method and SSO of the

¹¹⁶ *See id.* at 5.

¹¹⁷ *See id.* at 6-7.

¹¹⁸ *See id.* at 7.

¹¹⁹ *See id.*

¹²⁰ *See id.* at 1352.

¹²¹ *See Oracle*, 750 F.3d at 1359.

¹²² *Id.* at 1377.

¹²³ *Id.* at 1359.

¹²⁴ *See* 28 U.S.C. § 1295(a)(1) (2015); *see also Oracle*, 750 F.3d at 1353.

¹²⁵ The standard of review was de novo because "[f]air use is a mixed question of law and fact.' Thus, while . . . 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." *Oracle*, 750 F.3d at 1373 (citations omitted).

thirty-seven APIs were subject to copyright protection.¹²⁶ Because the district court never did any fact-finding on the fair use defense, the Federal Circuit remanded the case back down to the district court to decide on the fair use of the thirty-seven APIs according to the Federal Circuit's ruling that the APIs are copyrightable.¹²⁷ Google filed a petition for certiorari to the Supreme Court, but was ultimately denied and forced to return to the District Court for the Northern District of California for a new trial.¹²⁸

C. District Court Results from the Remanded Oracle v. Google Case

Upon remand, Google once again raised a fair use defense against Oracle's copyright infringement claims, for which Oracle was seeking \$8.8 billion in damages from profits Oracle claimed Google made from Android.¹²⁹ The next section¹³⁰ discusses the district court's interpretations of each of the fair use elements laid out above.¹³¹ To the entertainment of those in attendance,¹³² each

¹²⁶ See *Oracle*, 750 F.3d at 1353, 1367.

¹²⁷ *Id.* at 1377.

¹²⁸ See *Google, Inc. v. Oracle Am., Inc.*, 2015 U.S. LEXIS 4472 (2015).

¹²⁹ Joe Mullin, *Second Oracle v. Google Trial Could Lead to Huge Headaches for Developers*, ARSTECHNICA (May 8, 2016, 7:00 AM), <http://arstechnica.com/tech-policy/2016/05/round-2-of-oracle-v-google-is-an-unpredictable-trial-over-api-fair-use>.

¹³⁰ See *infra* Part V. Section A.1.

¹³¹ See *supra* Part II.

¹³² See *Twitter Collection of @SarahJeong, @Xor & @Swiftstories Oracle vs Google Coverage*, API EVANGELIST (May 10, 2016), <http://apievangelist.com/2016/05/10/twitter-collection-of-sarahjeong-xor-swiftstories-oracle-vs-google-coverage>. Tweets from the audience include: "it's fascinating that Oracle's lawyer did the whole 'WELL IT'S NOT LIKE IT'S A PAINTING' criticism of the fair use defense (9:09 AM, <https://twitter.com/sarahjeong/status/730067244365946881>) . . . but are very insistent that Java APIs are a creative work (9:09 AM <https://twitter.com/sarahjeong/status/730067349928169473>) . . . buddy, it's not like it's a painting." (9:10 AM, <https://twitter.com/sarahjeong/status/730067453326168065>); "You like Angry Birds, right? That's what Oracle hates. Angry Birds, and freedom." (9:46 AM, <https://twitter.com/sarahjeong/status/730076505221439489>); "I think both sides have brought LITERAL PHYSICAL FILE CABINETS as visual aids to explain

side presented their case, and the jury returned a verdict in favor of Google, concluding that Google's use of the thirty-seven Java APIs and their SSOs constituted fair use.¹³³

Oracle has filed an appeal of the verdict to the Federal Circuit.¹³⁴ As noted above,¹³⁵ the next part will attempt to analyze two potentially significant outcomes of that appeal and the consequences that could result from each.

V. RULING IMPLICATIONS

This part addresses the implications of a Federal Circuit ruling by analyzing (A) potential consequences of an affirmation of the District Court's ruling on Google's fair use and alternatively, (B) potential consequences of a reversal of the District Court's ruling on Google's fair use of the Java APIs.

A. *What Affirming the Current Ruling Could Mean*

To understand what an affirmation of the jury verdict by the Federal Circuit could mean, it is important to address the outcome of the jury trial. As mentioned above, the jury found in favor of

what an API is (9:56 AM, <https://twitter.com/sarahjeong/status/730079061293195264>) . . . he literally opened the top drawer and pulled something out and waved it around to explain (9:57 AM, <https://twitter.com/sarahjeong/status/730079061293195264>)” Sarah Jeong, @sarahjeong, TWITTER (May 10, 2016); “Oracle attorney is in for the challenge of his legal career: he is now trying to convince the jury that Java Is Good.” (8:30 AM, <https://twitter.com/xor/status/730057511974244352>); “Oracle attorney just held up a literal painting that has some Java code written on it as a REAL example of fair use” (9:00 AM, <https://twitter.com/xor/status/730064881852907520>); “Judge Alsup just had to read a dozen Android version names in the jury instructions. Some day you gotta hear a federal judge list desserts.” (10:23 AM, <https://twitter.com/xor/status/730085890547023872>) Parker Higgins, @xor, TWITTER (May 10, 2016).

¹³³ See Final Judgment, *supra* note 4.

¹³⁴ Notice of Appeal to the Fed. Cir. by Oracle Am., Inc, No. 3:10-cv-03561, entry # 2071, filed on 10/26/2016, <https://assets.documentcloud.org/documents/3183383/e159d3de-ee97-4d07-92df-db808e4a43c5.txt>.

¹³⁵ See *supra* Part I.

Google.¹³⁶ Jury deliberations are secret, but looking at how Judge Alsup presented the case and the law to the jury provides an understanding of the current ruling. This section proceeds by (1) describing more specifically the current interpretations of the statutory factors as charged to the jury and (2) why these interpretations should be affirmed.

I. The Statutory Factors as Charged to the Jury

While it is impossible to know exactly how the jury weighed each fair use statutory factor, Judge Alsup's¹³⁷ instructions to the jury in the district court show how he felt the four factors should be weighed.¹³⁸ According to Judge Alsup, the fourth statutory factor (market effect) is the "single most important statutory factor,"¹³⁹ followed by the first (purpose) and then third (amount) factors.¹⁴⁰ This is yet again a different order from the cases above.¹⁴¹ However, what is significant about Judge Alsup's interpretation of each of the factors is that they appear to be a liberal and lenient application of copyright law for software development¹⁴² that are more in line with the goal of the fair use doctrine to foster innovation.¹⁴³ This application is something the Federal Circuit has the potential to reinforce and continue, if not expand. Additionally, the Federal Circuit could also set a trend for the general order of importance for each of the four statutory factors.

¹³⁶ See *supra* Part IV. Section C.

¹³⁷ Judge Alsup is the District Court judge who presided over the initial *Oracle v. Google* case as well as the remanded case.

¹³⁸ See Final Charge, *supra* note 9, at 12-19.

¹³⁹ *Id.* at 18.

¹⁴⁰ See *id.*

¹⁴¹ See *supra* Part II.

¹⁴² See Final Charge, *supra* note 9, at 18; see also *infra* Part V. Section A.2.

¹⁴³ See Richard Stim, *What Is Fair Use?*, STANFORD UNIVERSITY LIBRARIES: COPYRIGHT & FAIR USE, <http://fairuse.stanford.edu/overview/fair-use/what-is-fair-use/> (last visited Nov. 18, 2016) (noting that the rationale behind the fair use doctrine is to allow the public to benefit from the improvement of the copyrighted work).

Like the Supreme Court in *Harper & Row*, Alsup believed the fourth factor to be the “single most important statutory factor” but noted it is “not necessarily dispositive” and “must be weighed with all other factors.”¹⁴⁴ The analysis must consider the extent to which the accused work can be a substitute or replacement for the copyrighted work, as well as the likelihood of future market harm.¹⁴⁵ For this factor to weigh in favor of fair use, the accused work should not “materially impair[] the marketability or value of the copyrighted work.”¹⁴⁶

For the first factor (purpose), Judge Alsup appeared to focus most on the transformative nature of the use rather than if the use was commercial.¹⁴⁷ He conceded that whether the use was commercial is also an element of the factor, but noted all parties agreed the purpose of the use was commercial.¹⁴⁸ Judge Alsup was very liberal in his definition of “transformative.”¹⁴⁹ While he stated the use must add something new, Judge Alsup defined the term transformative to include using the copyrighted work in a different context without being required to change the elements in any other way.¹⁵⁰

Again, like the Supreme Court in *Harper & Row*, for the third factor (amount), Judge Alsup put emphasis on the qualitative amount of copyrighted work used rather than the quantitative

¹⁴⁴ See Final Charge, *supra* note 9, at 18.

¹⁴⁵ *Id.* at 18-19.

¹⁴⁶ *Id.* at 18.

¹⁴⁷ See *id.* at 14.

¹⁴⁸ *Id.*

¹⁴⁹ See *id.* at 13-14.

¹⁵⁰ *Id.* at 13. A more conservative definition of “transformation” might require some actual change to the work, rather than allowing for the work to be used in a different context. Merriam-Webster defines transformations as “a complete or major change in . . . something’s appearance, form, etc.” See *Transformation*, MERRIAM-WEBSTER DICTIONARY, <https://www.merriam-webster.com/dictionary/transformation> (last visited Nov. 18, 2016). Because Judge Alsup says a transformation only requires a change in context his interpretation is more lenient and liberal. Final Charge, *supra* note 8, at 13 (“A use is transformative if it adds something new, with a further purpose or different character . . .”).

amount, even concluding “the total number of lines [of code] in Android is irrelevant.”¹⁵¹ He went further, allowing some wiggle room for qualitative copying, but only to the extent that it would facilitate a transformative use and that the extent related to the purpose and character of use in the first statutory factor (purpose).¹⁵²

While he appeared to emphasize least on the second statutory factor (nature), Judge Alsup followed an interpretation similar to that of the Court of Appeals of the Ninth Circuit.¹⁵³ The jury instructions insinuated that the jury should apply the AFC test, which implied juror analysis should hinge on whether the work is creative or functional, with functional weighing in favor of fair use.¹⁵⁴

2. *Why the Charged Interpretations Should Be Affirmed*

The Federal Circuit should affirm Judge Alsup’s interpretation and application of the statutory factors for two main reasons. First, under Judge Alsup’s interpretations set out in the jury instructions, the law would be more lenient and favorable to programmers, focusing more on a functional rather than formal analysis. A functional analysis allows for flexibility in characterization—accounting for realities in the field—whereas a formal analysis is more likely to adhere to the letter of the law. Computer programming is dynamic in nature, as evidenced by the number of open source projects running on sites like GitHub.¹⁵⁵ The

¹⁵¹ See Final Charge, *supra* note 9, at 17.

¹⁵² *Id.*

¹⁵³ *Sega Enters. V. Accolade, Inc.*, 977 F.2d 1510 (9th 1992).

¹⁵⁴ See Final Charge, *supra* note 9, at 15-16.

¹⁵⁵ See *Open Source*, GITHUB, <https://github.com/open-source> (last visited Oct 25, 2016); see also Sarah Jeong, *Why the Very Silly Oracle v. Google Trial Actually Matters*, MOTHERBOARD (May 25, 2016, 6:10 PM), <http://motherboard.vice.com/read/why-the-very-silly-oracle-v-google-trial-actually-matters> (implying a functional analysis better aligns with Google’s—and other developers’—reasons for using APIs, stating “APIs are highly functional” and that “[a] screwdriver looks like a screwdriver because it *has to*,” using the screwdriver analogy to explain why the Java APIs in the Android OS look the way they do: they have to) (emphasis added); *infra* Part V. Section B.1.

instructions appear to demonstrate that Google’s use of the Java APIs aligns with the collaborative, functional nature of software development.¹⁵⁶ Second, initiating a trend of consistency could lead to results that are more predictable and subsequently, a better understanding of how copyright law should be applied to software code.

For this case, Google argued that its actions had no bearing on Oracle’s market for Java, but rather, that any failures were “attributable to [Oracle’s] own actions or inaction”¹⁵⁷ On multiple occasions, Oracle and Sun both failed to adopt Java for mobile devices—at times due to insufficient functionality—even utilizing the same APIs at issue in this case.¹⁵⁸ Additionally, Google argued that the evidence showed that not only did using its own implementing code for the thirty-seven Java APIs change elements of the copyrighted work, but it also put the APIs into a different context.¹⁵⁹ This was transformative, even according to Oracle’s own employees.¹⁶⁰

Furthermore, Google noted that the nature of its use of the declaring code and SSO was mainly functional, stating “[the APIs’] predominant purpose is simply to allow access to the pre-written code in the API packages and thereby allow for the practical use of the Java programming language.”¹⁶¹ Demonstrating the functionality of the code, Google stated that “[the thirty-seven] Java . . . API packages and their organization are . . . oftentimes needed to make practical use of the Java programming language.”¹⁶² To give further evidence in favor of the nature of the copyrighted work, Google addressed the fact that Sun made Java available under a free open source license,¹⁶³ which “permitted

¹⁵⁶ See *infra* Part V. Section B.

¹⁵⁷ See Google’s Trial Brief, *supra* note 80, at 1, 10.

¹⁵⁸ *Id.*

¹⁵⁹ See *id.* at 10.

¹⁶⁰ *Id.*

¹⁶¹ See *id.*

¹⁶² *Id.* at 8.

¹⁶³ Google’s Trial Brief, *supra* note 80, at 8. In this instance, Java was released as part of a platform called “OpenJDK.” See Google’s Trial Brief,

licensees to subset or superset the Java . . . API packages, thereby enabling exactly the sort of fragmentation that [Oracle] claims to be damaged by.”¹⁶⁴

Google’s arguments appear to fit well into the statutory interpretations laid out by Judge Alsup in his instructions to the jury.¹⁶⁵ For factor four (market effect), Google and Judge Alsup both have a common understanding that the market effect relates to the manner in which the copyrighted material was used.¹⁶⁶ Google used the Java APIs for a mobile OS, and Oracle did not have a mobile OS,¹⁶⁷ thus there was no market for Google’s use to affect. This is a pragmatic approach to looking at market effect, as opposed to Oracle’s formal approach.¹⁶⁸ In calculating damages for harm, Oracle attempted to include everything the Java code was involved in.¹⁶⁹

supra note 79, at 10. OpenJDK, and by extension the Java code included in the platform, was released under the GPLv2 license with a ClassPath Exception. *See OpenJDK FAQ*, OPENJDK (Dec. 18, 2010), <http://www.openjdk.java.net/faq>. GPLv2 permits the licensee to use, copy, modify, and distribute the program’s source code provided, among other requirements, the licensee releases the source code upon distribution. *See GNU General Public License, version 2, with the Classpath Exception*, OPENJDK, <http://www.openjdk.java.net/legal/gplv2+ce.html> (last visited Oct. 25, 2016).

¹⁶⁴ *Id.*

¹⁶⁵ *See* Google’s Trial Brief, *supra* note 80, at 9-10; *see also* Final Charge, *supra* note 9, at 14-19.

¹⁶⁶ *See* Google’s Trial Brief, *supra* note 80, at 10; *see also* Final Charge, *supra* note 9, at 18-19.

¹⁶⁷ *See* Google’s Trial Brief, *supra* note 80, at 10.

¹⁶⁸ *See id.* at 12-14.

¹⁶⁹ *See id.* (describing Mr. Malackowski’s—Oracle’s expert—damage calculations). Mr. Malackowski’s damage calculations totaled \$8.8 billion for everything he felt Google’s use of the Java APIs affected, including: attracting developers who continued to develop more apps; apps that drew in customers; apps that inspired searches on Android devices, which generated “Android-related revenue”; sales of hardware that run Android; sales of apps on Android devices; sales of “digital content that Android device uses can *download*” (emphasis added); and advertising revenues generated from web searches performed on an Android phone. *See id.* at 12. This calculation was done without any supporting evidence or analysis demonstrating that the thirty-seven APIs in question caused or enabled any of the profit streams above. *See id.* Mr.

For factor one (purpose), again Judge Alsup and Google were on the same page, honing in on the transformative nature, rather than the commercial nature of the use.¹⁷⁰ While it behooved Google to direct attention away from the commercial nature of the use, an interpretation of factor one that is more focused on the transformative nature could reward innovation, permitting the public to benefit from technological advancements that may utilize building blocks like the declaring codes of the Java APIs. After all, it is with the use of those Java APIs that Google was able to develop Android, which currently dominates the smartphone OS market.¹⁷¹

The pattern of parallel thinking between Judge Alsup and Google continued with factor two, which addressed the nature of the work.¹⁷² When certain APIs and SSOs are required for practical use, it would inhibit progress to prevent usage of those necessary elements. Additionally, when the code is released to the general public for use following a relatively unrestricted open-source license, the code, and subsequently Google's use of that code, comports with the collaborative nature of the software industry and the functional interpretation set out by Judge Alsup.

Google, Judge Alsup, and the jury all appear to understand that the purpose of a fair use defense is to promote technological innovation. There is, however, no guarantee that the Federal Circuit will set any standards according to Judge Alsup's interpretations of the statutory factors. There is not even a guarantee that the Federal Circuit will not reverse the jury verdict.

Malackowski even argued that requiring that analysis “would be impermissible and irrelevant.” Google's Trial Brief, *supra* note 80, at 12. Mr. Malackowski's damage calculations demonstrate a formalistic analysis of the fourth statutory factor (market effect), which goes against the idea of innovation through collaboration demonstrated by many software developers.

¹⁷⁰ See *id.* at 9-10; see also Final Charge, *supra* note 9, at 14.

¹⁷¹ See *Smartphone OS Market Share, 2016 Q2*, IDC, (Aug. 2016) www.idc.com/prodserv/smartphone-os-market-share.jsp (“Android dominated the market with an 87.6% share in 2016Q2.”).

¹⁷² See Google's Trial Brief, *supra* note 80, at 10; see also Final Charge to the Jury, *supra* note 9, at 15-17.

Reversing the jury verdict could have detrimental effects for both copyright law and the fair use test, as well as for the software field.

B. What Reversing the Verdict Could Mean

Having addressed the district court's interpretations of the statutory factors and why the Federal Circuit should affirm the jury's verdict,¹⁷³ it is necessary to now examine the potential consequences of the alternative result of Oracle's appeal. This section theorizes possible outcomes of a reversal by the Federal Circuit of the district court's verdict, focusing on (1) how it may affect copyright law and (2) how it may affect the software industry.

1. Effect on Copyright Law

It is necessary to first address the precedential impact of a reversal. The Federal Circuit's decision may end up having a low precedential impact because the Federal Circuit is not the court that would normally hear a copyright case.¹⁷⁴ As stated above,¹⁷⁵ this case only went to the Federal Circuit because there was a patent infringement claim initially involved, and the Federal Circuit has exclusive jurisdiction for appeals of patent cases.¹⁷⁶ Additionally, the Federal Circuit applied Ninth Circuit law.¹⁷⁷ Because the Federal Circuit is neither the Ninth Circuit nor the Supreme Court, it does not have the final say on how to interpret Ninth Circuit law.

There may be minimal change in how the law is actually interpreted and applied. To start with, the outcome of the appeal

¹⁷³ See *supra* Part V. Section A.

¹⁷⁴ Unless brought before the United States Court of Federal Claims—which does get appealed to the Federal Circuit, see 28 U.S.C. § 1295(a)(3)—a copyright infringement claim brought in federal district court would follow like any other case brought in federal district court: an appeal to the respective circuit court, potentially followed by an appeal to the Supreme Court. See *generally*, *Sega Enters. V. Accolade, Inc.*, 977 F.2d 1510 (9th 1992); *Harper & Row, Publr. v. Nation Enters.*, 471 U.S. 539 (1985); *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417 (1984); *Mullin, supra* note 129.

¹⁷⁵ See *supra* Part IV. Section B.

¹⁷⁶ See 28 U.S.C. § 1295(a)(1) (2015).

¹⁷⁷ See *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1353 (Fed. Cir. 2014).

will not affect the copyrightability of the literal and non-literal elements of APIs because the Federal Circuit's determination on that matter stands and will not be overturned.¹⁷⁸ Because APIs will still be copyrightable until a court with binding authority (the Ninth Circuit or the Supreme Court in this case) says otherwise, that is the way copyrightability must be interpreted.¹⁷⁹ Due to the potentially low binding or persuasive authority of the Federal Circuit, a case before the Ninth Circuit on the copyrightability of APIs could overturn the Federal Circuit's decision with respect to the Ninth Circuit, rendering any discussion stemming from this case relatively moot.

Additionally, the scope of the case is narrow—focusing only on whether Google's use of the declaring code and SSO of the Java APIs was fair.¹⁸⁰ Because a fair use defense is fact dependent, a reversal in this case may simply mean that under these particular facts, the Federal Circuit did not believe that Google's use of the Java APIs was fair. A reversal would not inherently invalidate another entity's use of the Java APIs, or any APIs in general.

Furthermore, there may not be any increased risk of litigation. Again, a reversal would not be saying that no actor would have a fair use defense for use of APIs, only that Google did not. Actors copying implementing code, declaring code, or SSOs of APIs could still be subject to copyright infringement claims, just as they could be now. Larger, more commercialized actors will still need to tread carefully with regard to their use of such code, and may only have a marginally larger target on their back due to a potentially perceived boost this case could give to a plaintiff's complaint. On the other hand, smaller actors and average users may not actually feel any additional pressure from larger companies. This is not to say that the smaller actors would not face any litigation risk, but rather that they may not be at any greater

¹⁷⁸ The copyrightability stands because it was not at issue in this case, *see* Final Charge, *supra* note 9, at 11, thus it is not reviewable by the Federal Circuit.

¹⁷⁹ *See Oracle*, 750 F.3d at 1358-59.

¹⁸⁰ *See* Final Charge, *supra* note 9, at 11.

risk from larger companies than they already are because the cost of litigation may outweigh the recovery.

To the extent its opinion would be binding or persuasive, the Federal Circuit could have the potential to negatively shape copyright law as it applies to software. Regardless of whether the Federal Circuit reverses the jury verdict or not, it will likely lay out an interpretation of the four statutory factors of fair use doctrine, as appellate courts generally do when presented with a test. It is possible the Federal Circuit's interpretations could line up with those of Judge Alsup, but find that the evidence in the record points in a different direction. However, it is equally likely that the Federal Circuit could focus on and give different weight to other factors that do not line up with those emphasized by Judge Alsup.¹⁸¹

On the other hand, a possible consequence of the Federal Circuit developing its own factor interpretations is that it could restrict what constitutes fair use, strengthening copyright law. This could result in analytically bad law. Concerning technology, analytically bad law is law that would seem to comport with statutory language, but is largely impractical in real world application.¹⁸² It would contradict how the field it is governing actually functions.¹⁸³

For example, a shift from analyzing the first statutory factor based on a transformative nature to focusing solely on whether a use is commercial could have a strong negative impact on innovation. In the technology world, the big name companies are commonly the ones with the most money to spend on advancements and innovations. They have the most money, in part,

¹⁸¹ See *supra* Part V. Section A.1.

¹⁸² See Sarah Jeong, *In Oracle v. Google, a Nerd Subculture Is on Trial*, MOTHERBOARD (May 12, 2016, 2:46 PM), <http://motherboard.vice.com/read/in-google-v-oracle-the-nerds-are-getting-owned> (discussing how the issue is being decided by people who do not understand APIs).

¹⁸³ See Jonathan Band, *The Federal Circuit's Poorly Reasoned Decision In Oracle v. Google*, PROJECT DISCO (May 9, 2014), http://www.project-disco.org/intellectual-property/050914-the-federal-circuits-poorly-reasoned-decision-in-oracle-v-google/#.V_AZEsmXrh4.

because they are commercializing at least some of their work product. Focusing on whether a use was commercial rather than how transformative that use may have been disregards potential benefits the public might gain from the improvement. It is relatively easy to classify work product coming out of a larger company like Google as commercial, because they are a business, and have shareholders and investors to whom they answer. In the 2012 trial, Google stated that “[n]obody is claiming that Google created Android as part of a charitable mission. The evidence is pretty clear that they created it to provide a platform on which other Google products could do better.”¹⁸⁴ However, there are technological benefits to such development, as demonstrated by Android’s market dominance.¹⁸⁵ Performing a more formal analysis that focuses on the commercial aspect as Oracle implored,¹⁸⁶ rather than looking at the transformative nature of the disputed work, provides an easily checked box against the defendant regardless of societal gains.

It is difficult to predict how the Federal Circuit may interpret the different statutory factors upon review or if that ruling will have any influence on other courts. However, to the extent the Federal Circuit offers different interpretations and other courts are influenced by the Federal Circuit’s decision, shifting interpretations toward example above could lead to real world consequences, such as a debilitating effect on the software industry.

2. *Effect on the Software Industry*

Other than the law becoming outdated and more stringent, consequences of a reversal of the jury verdict could include a debilitated software industry and diminished innovation. When the Federal Circuit first decided that APIs were copyrightable, the

¹⁸⁴ See Oracle’s Trial Brief at 3, Oracle Am., Inc. v. Google Inc., No. C-10-03561 WHA (N.D. Cal. June 8, 2016).

¹⁸⁵ See *Smartphone OS Market Share*, *supra* note 171 (“Android dominated the market with an 87.6% share in 2016Q2.”).

¹⁸⁶ See Oracle’s Trial Brief, *supra* note 184, at 3.

software industry was confused.¹⁸⁷ The notion of copyrighting non-literal elements and literal elements like method declaration did not correspond with the general expectations of developers.¹⁸⁸ Most developers wrote code for others in the industry to use, furthering collaboration and technological progress.¹⁸⁹ Suddenly, the Federal Circuit had put a potential limitation on that collaboration and progress.¹⁹⁰ A reversal by the Federal Circuit of the jury verdict on fair use could magnify that confusion and uncertainty.

In some ways, the outcome of an appeal to the Federal Circuit will have little to no bearing on the software world. Damage was done with the first appeal to the Federal Circuit, in which the court held declaring code and SSOs of APIs were copyrightable.¹⁹¹ As stated above, an appeal of the fair use defense is not going to change that fact, so developers will still have to be cognizant of that for now. Additionally, as stated above, the independent developers and collaborative teams are unlikely to feel any litigation pressure, at least from powerhouses like Oracle, who are only seeking big damages.

However, a reversal may, to the detriment of independent and smaller developers, provide an environment for bad-faith

¹⁸⁷ See e.g., Jeong, *supra* note 182 (“Google’s nerd witnesses are hamstrung on their ability to explain the motivations behind their actions, because . . . no one thought the Java APIs were copyrightable.”); Jared Newman, *Here’s The Scariest Thing About The Oracle-Google Software Copyright Battle*, FAST COMPANY (July 2, 2015, 10:10 AM), <https://www.fastcompany.com/3048218/app-economy/heres-the-scariest-thing-about-the-oracle-google-software-copyright-battle> (addressing the uncertainty the result could mean for future developing); Jeong, *supra* note 155 (describing that copyrighting APIs does not match up with typical use of APIs, quoting “A screwdriver looks like a screwdriver because it has to. A fair use trial over APIs is as sensible as a fair use trial over screwdriver heads.”); Band, *supra* note 183 (noting that the Federal Circuit relied on discredited case law from the Third Circuit, misconstruing and ignoring binding Ninth Circuit law as well as persuasive law from other circuits and court systems).

¹⁸⁸ See Jeong, *supra* note 155.

¹⁸⁹ See *id.*

¹⁹⁰ See *id.*

¹⁹¹ Oracle Am., Inc. v. Google Inc., 750 F.3d 1339, 1358-59 (Fed. Cir. 2014).

opportunists to take advantage of the industry.¹⁹² This environment would most likely be the result of a more formalistic interpretation of the statutory factors for a fair use defense. As mentioned above, a formalistic approach can, in some cases, provide easily checked boxes for plaintiffs. While it may not be certain that a decision by the Federal Circuit would have any precedential influence to a court, that precedent is not necessarily required for those bad-faith opportunists to be successful; the presence of those easy-to-check boxes may be all the ammunition needed. A potential market for copyright trolls¹⁹³ and licensing businesses could grow from weakened fair use defenses and strict enforcement of copyrights of things like declaring codes and SSOs.¹⁹⁴

These licensing businesses would most likely behave like patent trolls, which are non-practicing entities that buy up patent portfolios, and as with copyright trolls, their sole aim is to find people to sue for infringement and make money on subsequent settlements.¹⁹⁵ Unwitting independent developers and users, who might not even know they needed a license, could be subjected to strong-armed licensing agreements to avoid being taken to court. If those users or developers are not legally sophisticated they may not know that the Federal Circuit's opinion could potentially have little weight at trial. At the same time, regardless of sophistication and being taken advantage of, they may want to avoid the costs, time, and effort associated with trials.

To shield themselves and downstream recipients from copyright trolls, independent and smaller developers who intended their programs and modifications to be freely distributed and used may need to do extra work. Developers who want their programs

¹⁹² See Jorge Espinosa, *Don't Underestimate The Copyright Trolls*, LAW360 (Jan. 22, 2015, 10:30 AM), <http://www.law360.com/articles/612235/don-t-underestimate-the-copyright-trolls>.

¹⁹³ See Espinosa, *supra* note 192 (“These alleged trolls . . . threaten copyright lawsuits in exchange for a quick settlement.”).

¹⁹⁴ See *Copyright Trolls*, ELECTRONIC FRONTIER FOUNDATION, <https://www EFF.org/issues/copyright-trolls> (last visited Nov. 11, 2016); see also Espinosa, *supra* note 192.

¹⁹⁵ See Espinosa, *supra* note 192.

to be freely used would need to go back through their work and ensure that any technology, such as declaring code and SSOs of APIs, are wholly their own, or are appropriately licensed under a proprietary or open-source license.¹⁹⁶ Additionally, the developers would need to publicly announce inclusion of the licenses in their code so the downstream users would know of the license requirements.¹⁹⁷ This can be an insurmountable task. For example, one site that hosts open-source programming projects is GitHub.¹⁹⁸ GitHub allows for free collaboration on any number of projects, and there are millions of projects hosted.¹⁹⁹ A single developer could have contributed to any number of projects, which would require a search through each line of code in each project to verify the correct license is attached. Such an undertaking could be impracticable.

Additionally, the work-around requirements for those software programs that utilize Java APIs could be substantially detrimental to interoperability.²⁰⁰ New methods may need to be developed and redistributed to get those utilizing Java APIs back to the level of compatibility and efficiency at which development is currently operating.²⁰¹ Java took at least four years from its conception to break through into the public spotlight and had been continuing to

¹⁹⁶ See *Licenses & Standards*, OPEN SOURCE INITIATIVE, <https://opensource.org/licenses> (last visited Oct. 1, 2016).

¹⁹⁷ See *id.*

¹⁹⁸ See GITHUB, <https://github.com> (last visited Oct. 25, 2016).

¹⁹⁹ See *Open Source*, GITHUB, <https://github.com/open-source> (last visited Oct 25, 2016).

²⁰⁰ See *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1368 (Fed. Cir. 2014) (“In characterizing the SSO of the Java API packages as a ‘method of operation,’ the district court explain that ‘duplication of the command structure is necessary for interoperability.’ The court found that, ‘in order for at least some of the pre-Android Java code to run on Android, Google was required to provide the same . . . command system using the same names with the same “taxonomy” and with the same functional specifications.’” (quoting *Oracle Am., Inc. v. Google Inc.*, 872 F. Supp. 2d 974, 977 (N.D. Cal. 2012)) (internal citations omitted)).

²⁰¹ See Band, *supra* note 183; see also Jeong, *supra* note 155 (discussing how common Google’s reimplementation of APIs is, stating “no one has clean hands,” and exemplifying Apache Harmony, IBM, and Oracle).

grow for a total of fifteen years by the time Google began to develop the Android OS.²⁰² While technological growth has been increasing annually, there is still the potential for a heavy workload to return to the level of interoperability that Java provided.

VI. CONCLUSION

The current state of software copyright law is murky, but the Federal Circuit has an opportunity to clear things up. There are potential benefits should the Federal Circuit affirm the jury verdict. An affirmation could result in continued, if not more, lenient interpretations of the statutory factors, employing a functional—rather than formalistic—analysis. This would create copyright law that focuses more on permitting innovation. Additionally, an affirmation could initiate a potential order of importance of those statutory factors. Because of these benefits, as well as the evidence presented by Google, the Federal Circuit should affirm the District Court’s ruling.

A reversal, however, could go either way with respect to legal effects. There could be no major legal effects due to the potentially low precedential influence the Federal Circuit has on other courts. Although less likely, the decision could also be detrimental in prompting a stricter, more formalistic approach to fair use. Regardless of the precedent the Federal Circuit may or may not set for other courts, there still could be damaging effects to the software industry, where reality and practice can be independent of a court’s interpretation of certain laws. Copyright trolls, widespread uncertainty, and diminished innovation could befall the industry as a result of ambiguity created by a reversal of the district court’s holding. These are harms that should be recognized as being bigger than the case itself. Their potential impact on the industry should outweigh what Oracle hopes to gain in a victory over Google. The Federal Circuit has the opportunity to set a positive example, and it should take that opportunity, rather than handicapping an entire industry.

²⁰² See *Java Timeline*, ORACLE, <https://www.oracle.com.edgesuite.net/timeline/java/> (last visited Oct. 25, 2016).