

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

KEKER & VAN NEST LLP
ROBERT A. VAN NEST - # 84065
rvannest@kvn.com
CHRISTA M. ANDERSON - # 184325
canderson@kvn.com
DANIEL PURCELL - # 191424
dpurcell@kvn.com
633 Battery Street
San Francisco, CA 94111-1809
Telephone: 415 391 5400
Facsimile: 415 397 7188

KING & SPALDING LLP
DONALD F. ZIMMER, JR. - #112279
fzimmer@kslaw.com
CHERYL A. SABNIS - #224323
csabnis@kslaw.com
101 Second Street, Suite 2300
San Francisco, CA 94105
Tel: 415.318.1200
Fax: 415.318.1300

KING & SPALDING LLP
SCOTT T. WEINGAERTNER
(Pro Hac Vice)
sweingaertner@kslaw.com
ROBERT F. PERRY
rperry@kslaw.com
BRUCE W. BABER (Pro Hac Vice)
1185 Avenue of the Americas
New York, NY 10036
Tel: 212.556.2100
Fax: 212.556.2222

IAN C. BALLON - #141819
ballon@gtlaw.com
HEATHER MEEKER - #172148
meekerh@gtlaw.com
GREENBERG TRAURIG, LLP
1900 University Avenue
East Palo Alto, CA 94303
Tel: 650.328.8500
Fax: 650.328.8508

Attorneys for Defendant
GOOGLE INC.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,

Plaintiff,

v.

GOOGLE INC.,

Defendant.

Case No. 3:10-cv-03561 WHA

**GOOGLE INC.'S PROPOSED FINDINGS
OF FACT AND CONCLUSIONS OF LAW
REGARDING ISSUES OF FACT AND
LAW THAT MUST BE RESOLVED BY
THE COURT**

Dept.: Courtroom 8, 19th Floor
Judge: Hon. William Alsup

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

TABLE OF CONTENTS

	<u>Page</u>
I. FINDINGS OF FACT.....	1
A. Copyrightability	1
1. API Structure, Sequence, and Organization	1
2. API packages and their SSO provide vocabulary and functionality to the Java programming language.	4
3. Some of the 37 Java API packages, or their elements, are essential to use the Java language.....	6
4. The 37 Java API packages and their SSO provide functionality relied on by Java programmers.	7
5. In order to provide the functionality expected by Java language programmers, the SSO of the Java API packages must be the same.....	10
B. Equitable Defenses.....	12
1. Sun knew about and approved unlicensed, open source implementations of the Java API packages as long as the implementation did not use the Java brand.....	12
2. As early as 2005, Sun knew Google intended to implement Java API packages in Android, and Sun never told Google it needed a license to do so.....	14
3. After Google publicly announced Android, Sun congratulated Google and welcomed Google to the Java community.	16
4. After Google’s announcement of Android and release of the SDK, Sun continued to talk with Google and publicly support Android.	19
5. Google was aware of and relied on Sun’s public statements of approval and acts of support for the Android platform.....	21
6. Oracle initially encouraged Android and tried to partner with Google.....	23
II. CONCLUSIONS OF LAW	24
A. Copyrightability	24
1. Section 102(b) of the Copyright Act.....	24
2. Merger and <i>scenes a faire</i>	28
B. Equitable Defenses.....	31
1. Laches	31

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

2. Equitable Estoppel32
3. Implied License.....34
4. Waiver.....35

TABLE OF AUTHORITIES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Page(s)

Federal Cases

Apple Computer, Inc. v. Microsoft Corp.
35 F.3d 1435 (9th Cir. 1994) 25

Baker v. Selden
101 U.S. 99 (1879)..... 28

Bateman v. Mnemonics, Inc.
79 F.3d 1532 (11th Cir. 1996) 29, 30

Computer Assoc. Int’l, Inc. v. Altai, Inc.
982 F.2d 693 (2d Cir. 1992)..... 26, 29, 30

Danjaq LLC v. Sony Corp.
263 F.3d 942 (9th Cir. 2001) 31

Effects Associates, Inc. v. Cohen
908 F.2d 55 (9th Cir. 1990) 34

Feist Pubs., Inc. v. Rural Tele. Serv. Co., Inc.
499 U.S. 340 (1991)..... 28

Lotus Dev. Corp. v. Borland Int’l, Inc.
49 F.3d 807 (1st Cir. 1995),
aff’d by an equally divided court, 516 U.S. 233 (1996) 26, 28

Lulirama Ltd., Inc. v. Axxess Broad. Services, Inc.
128 F.3d 872 (5th Cir. 1997) 34

Mitel, Inc. v. Iqtel, Inc.
896 F. Supp. 1050 (D. Colo. 1995),
aff’d 124 F.3d 1366 (10th Cir. 1997)..... 26

Sega Enters. Ltd. v. Accolade, Inc.
977 F.2d 1510 (9th Cir. 1992) 26

Wang Laboratories, Inc. v. Mitsubishi Electronics America, Inc.
103 F.3d 1571 (Fed. Cir. 1997)..... 34

Other Cases

SAS Inst., Inc. v. World Programming Ltd.
Case C-406/10 (E.U. Ct. Justice May 2, 2012)..... 26, 28

Federal Statutes

17 U.S.C. § 102(b) 24, 25, 27, 28

1
2 **I. FINDINGS OF FACT**

3 **A. Copyrightability**

4 **1. API Structure, Sequence, and Organization**

5 1. When programmers write programs using the Java programming language, they
6 access libraries of prewritten code to perform programming functions.

7 Astrachan at RT 2203:19-25

8 2. An application programming interface (“API”) is the names, words, and the set of
9 rules that a programmer uses to communicate with a library and access the prewritten code
10 contained in that library.

11 Bloch at RT 763:21-24

12 Q. How would you define an API?

13 A. I would say that an API is -- the names, the words, and the set of
14 rules that the programmer uses to communicate with a library.

15 Mitchell at RT 1300:21-1301:6

16 Q. Okay. Now, a programmer writing in the Java language must use the
17 names `java.lang.Math.max` in order to invoke the prewritten code
18 that performs that `max` function, correct?

19 A. Yes.

20 Q. And, therefore, these names actually have a functional purpose,
21 right?

22 A. The names are used exactly as I described; to access the compiled
23 output from the prewritten code.

24 Mitchell at RT 1297:23-1298:7

25 Ellison at RT 290:11-14

26 Ellison at RT 308:22-309:4

27 Kurian at RT 364:3-10

28 3. The prewritten code in the Java class libraries is divided into units known as
methods, each of which provides a specified functionality.

Bloch at RT 763:21-764:15

4. The Java language requires that methods be grouped into “classes,” which are then
grouped into “packages.”

Mitchell at RT 1306:14-17

Q. Now, the methods are organized into classes and the classes are
organized into packages. That’s the organization of the Java
language, right?

A. That’s part of the organization.

Astrachan at RT 2183:12-20

Mitchell RT 1235:5-12

1 5. To the extent the “structure, sequence and organization” (“SSO”) of the API
2 packages have a “hierarchical structure,” that structure is dictated and required by the Java
3 programming language, which requires that the fully-qualified name of any API method,
4 constructor or other API element must consist of the package name, the class name, and the
5 method name.

6 Astrachan at RT 2187:18-2188:4

7 Q. Okay. And, for example, what are the attributes of the structure of
8 an API?

9 A. The structure of an API is something that we have talked about here.
10 It’s the – in Java API because we’re only speaking about Java, APIs
11 in other languages might be different. Package name, Class name,
12 Method name.

13 Q. And would you call that a hierarchical structure?

14 A. Yes, I think that’s reasonable to call that hierarchical.

15 Q. Is that hierarchical structure required by the Java language
16 specification, or is it not?

17 A. The language specification requires Packages, Classes, Methods.

18 Bloch at RT 769:23-770:9

19 Q. In the Java language, Dr. Bloch, what is it that determines the
20 organization of the code libraries that implement the APIs?

21 A. The names of the methods basically determine that, because names
22 in Java, they have three parts. It’s called a fully-qualified name. It
23 consists of the class -- actually, the package, the class, and then the
24 method or field. So the whole name of something might be
25 something like java.lang.math.cos. So it’s not math. It’s a package.
26 And then -- sorry java.lang is a package. Math is a class. And then
27 cos is the cosine function. And so the name determines the
28 organization.

 Bloch at RT 774:4-775:5

19 6. The means by which the SSO is stated in tangible form in the code must use and
20 follow strictly the rules of the Java programming language. This means that any possible
21 expression of the ideas that result from “design choices” made in designing an API or API
22 package—the ideas regarding which methods should be put in which classes, which classes
23 should be put in which packages, which classes should extend which other classes, and which
24 classes should implement which interfaces—is constrained by the rules of the Java language.

25 Astrachan at RT 2190:16-23

26 Q. And how many lines of code actually implement the characteristics
27 of a class that you want to put in your new API?

28 A. There is one line for every method. There is one line for every class.
And there is a line for the package that I specify. So I have a
Package line, a Class line -- and that Class line has all the inter-
dependencies about what is inherited -- and then a line for every
Method.

1 7. All of the elements of the SSO for any given class or method are expressed in
2 either a single line of code or a small number of lines of code (five or fewer) for that class or
3 method, which must be written in accordance with the requirements of the Java language.

4 Astrachan at RT 2189:22-2191:20
5 Reinhold at 2241:3-2243:11, 2244:19-2245:15
6 Mitchell at RT 2300:13-25

7 8. The SSO for an API or API package are a system for identifying the location of
8 the prewritten code in the library analogous to how street addresses are a system for identifying
9 the location of a given building.

10 Bloch at RT 772:17-24; 773:14-16

11 [A.] So think of it as, you know, your city, your state, and your street.
12 They are all part of your address. So, you know, the APIs here are
13 the addresses, and the package is the city. The class is the street.
14 And method is the house number, if you want. It's not a great
15 analogy, but you get the idea.

16 THE COURT: Yes, but where is the API part?

17 [A.]: All three of them are parts of it.

18 ***

19 But, you know, as I say, I think the address metaphor is pretty good.
20 Which is a more important part of your address, the street or the
21 city?

22 9. Similarly, the SSO for an API or API package represents a command structure to
23 access prewritten computer code contained in the Java class libraries.

24 Ellison at RT 289:8-9

25 [A.] The APIs are a command structure you give to the program.

26 Ellison at RT 290:8-12.

27 Kurian at RT 364:3-10

28 Schwartz 1959:12-1960:18

29 10. The SSO of an API or API package is not like a blueprint. While blueprints tell
30 you how to build something, the SSO of an API or API package does not tell you how write the
31 implementing code. The SSO of an API or API package provides only the functional
32 requirements describing what the thing you are building—the implementing code—must do.

33 Bloch at RT 769:4-22

34 Astrachan at RT 2150:18-2151:17

35 Reinhold at RT 2240:5-20

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

2. API packages and their SSO provide vocabulary and functionality to the Java programming language.

11. The Java programming language, without any API packages and class libraries, consists only of the language’s grammar or syntax without a vocabulary.

Reinhold at RT 673:2-20

Q. Dr. Reinhold, what is the Java programming language as a formal -- in as formal a way as you can describe it?

A. As formal a -- so the Java programming language, like any language, really, has two fundamental parts. There’s the syntax for the language, which is essentially the grammar. We all remember in grade school taking English sentences and decomposing them down into their separate parts. There is a grammar just like that for the Java programming language and, really, for any programming language. The other important part of the language specification is the semantics -- that is a set of rules -- in this case expressed in English prose in several hundred pages which specify, as precisely as possible, what every construct you can express in the language means. So if there is a piece of the grammar, for example, that has a command for a conditional test, if something is true then do something, then the language speculation specification, describes in excruciating detail what that kind of a statement is supposed to mean.

12. Because it consists only of grammar and no vocabulary, the Java programming language has almost no functionality without API packages and their class libraries.

Reinhold at RT 683:14-684:4

Q. Without any kind of class library at all -- I want you to assume no class library -- what kind of programming could you do with the Java programming language itself?

A. With no class library at all, you could do very little.

Reinhold at RT 707:18-21
Bloch at RT 782:9-14
Schmidt at RT 1477:2-13
Schwartz at RT 1960:4-8

13. Without API packages and class libraries, a Java program could not communicate with a computer monitor so a user could read the output, could not communicate with a printer so a user could print the output, and could only do computation with primitive data types.

Reinhold at RT 683:14-684:4
Reinhold at RT 707:22-708:18
Mitchell at RT 1277:18-1278:1

14. The API packages provide the names and vocabulary used in the Java programming language.

Bloch at RT 784:9-21

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Q. Is there some respect, Dr. Bloch, in which the names of the APIs are like words that are used in the Java programming language?

A. Yes. They are very similar. The idea is that in order to do anything with a language, you don't want to be confined to the small set of words that are the so-called key words of the language. In Java that means like "while" and "do" and "for." You want to be able to add your own vocabulary specific to what you want to do with the language. So if you're going to run the language, say, to run a cash register, then you want to be able to add a word to express putting money into the till, taking money out of the till, and those new words are APIs.

Mitchell at RT 1304:5-20 (from TX 3542, video clip of Mitchell Depo at 120:18-24; 121:1-10)

Q. Okay. And using those same, roughly, seven elements identified -- or however many you want to count -- elements identified in paragraph 3 of the reply report as being part of the specification, going back to paragraph 55 of your opposition report, what do you consider those elements as a group, as a specification, to be a written expression of?

THE WITNESS: I mean, in a sense, these elements here are like parts of speech. So nouns, verbs, adjectives, and so on. And using those parts of speech, package names, class and interface names, relationships between them, the methods associated with each class, methods associated with interface, whether a method is static or not, all of those things that are expressed in Java, it can't -- it -- you form an API using those parts of speech in the same way I'd write a technical paper using, you know, nouns, verbs, objects, so on.

Screven at RT 542:7-10, 564:15-565:2

Bloch at RT 766:14-25

15. By providing names and a vocabulary for the Java programming language, API packages and their SSO provide a system of expression in the Java programming language.

Bloch at RT 746:24-747:9

Q. And this was -- I remember when we talked earlier, Mr. Bloch, this - this phrase struck me as quite powerful in your presentation. "Code should read like prose." Do you see that on your presentation?

A. Yes.

Q. What were you driving at?

A. Well, writing a program is very much a creative process. And if you have good words to use, if the API gives you good words that really mean what it is that you're doing with the API, then once it comes time for you to take that API and write a program with it, the program will read like English text.

Bloch at RT 747:25-748:6

Q. The point is, if you look at this code example—can the jury see the code example?

A. On the screen, yes, sir.

A. All right. It reads almost like English because the API has provided a good vocabulary to express what it is that it is providing for the programmers.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

3. Some of the 37 Java API packages, or their elements, are essential to use the Java language.

16. Several classes and methods in the 37 API packages at issue in this case are literally essential to any use of the Java programming language.

Mitchell at RT 1274:16-24

Q. In other words, without any of java.lang, the API, the language really doesn't work; isn't that right, Dr. Mitchell?

A. There are a few things, such as Object, that are in here that I think are essential to the language.

Q. So without java.lang, the language doesn't work because java.lang has things in it that are essential to the language, right?

A. Yes. There are also some things that might not be so essential to the language.

Reinhold at RT 684:16-685:2, 679:12-21 (61 classes in the accused APIs required by the Java Language Specification)

Bloch at RT 776:12-777:7, 777:19-778:9, 779:13-780:18

17. The 60 or 61 classes required to implement the Java language specification and make any use of the Java language reference over 170 other classes consisting of over 2,000 public methods and fields spread across ten packages.

Bloch at RT 779:13-780:18
TX 610.2
TX 984

18. The First Edition of the Java Language Specification included a specification for the java.lang, java.io, and java.util API packages.

TX 2564 at 23 ("This book attempts a complete specification of the syntax and semantics of the Java language and the core packages java.lang, java.io and java.util of its Application Programming Interface.")
Bloch at RT 781:11-24

19. The First Edition of the Java Language Specification stated that java.lang, java.io, and java.util packages "must be included in all general purpose Java systems."

TX 2564 at 31 ("Chapter 20 through 22 are the reference manual for the core of the standard Java Application Programming Interface. These packages must be included in all general purpose Java systems.")

20. Volume 1 of the Java Application Programming Interface described the java.lang, java.io, and java.util packages as "the foundation of the Java language" and as "the general purpose libraries fundamental to every Java program."

TX 980 at 528 ("Volume I, Core packages, describes the libraries that are the foundation of the Java language. These libraries include java.lang,

1 java.io, java.util, and java.net. These are the general purpose libraries
2 fundamental to every Java program.”)

3 **4. The 37 Java API packages and their SSO provide functionality relied**
4 **on by Java programmers.**

5 21. When Sun released the Java programming language, its goal was to encourage
6 widespread use of the language by encouraging as many people as possible to use the language.

7 Schwartz at RT 1957:24-1958:4

8 Q. And did the company take steps to promote widespread use of the
9 Java language?

10 A. The company, Sun Microsystems, worked as hard as we could to
11 open the market and using Java, and the distribution of Java and the
12 technologies behind Java to open that market, absolutely.

13 Schmidt at RT 1474:24-1475:10

14 22. Because the functionality provided by the API packages (as accessed through the
15 packages’ SSO) was necessary to make any meaningful use of the Java programming language,
16 Sun promoted the free use of the Java API packages developed as part of the free use of the Java
17 programming language.

18 Schwartz at RT 1961:13-19

19 Q. Mr. Schwartz, did Sun promote the Java language APIs along with
20 the language?

21 A. Absolutely. We had to, if you wanted to see that language broadly
22 accepted. So it’s insufficient to just give you a language because
23 what do you do with it? I mean, how do you now write an
24 application?

25 Schwartz at RT 1962:2-9

26 Q. So were the APIs simply marketed along with the language? In
27 other words, free and available for everyone?

28 A. Yes. Absolutely. We talked about open APIs, and then you
compete on implementations. And what that means is we all had the
same set of APIs, but we would then create products, the virtual
machine specifically or the technology that underlies the language,
to go off and perform – I’m doing a bad job of explaining.

Schmidt at RT 1477:2-1478-9

23 23. Sun did not consider the Java API packages or their SSO proprietary, and it
24 worked hard to dispel any suggestion that the API packages were proprietary.

25 Schwartz at RT 1966:1-12

26 Q. So, Mr. Schwartz, was there ever a time during your tenure at Sun,
27 all the way up to 2010, I believe you said, where the APIs were
28 considered – the Java APIs were considered proprietary or protected
by Sun?

A. No. And to the extent that anybody made that claim, we would have
worked very hard to say that’s not true. These are open APIs. We
want to bring in as many people as possible because if we did, we

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

can bring them together. Now they have added to the Java Community. Our market opportunity got that much bigger because more people were a part of the community. We wanted to basically build the biggest tent and invite as many people as possible.
Schwartz at RT 2003:7-20

24. Part of Sun’s active promotion of the Java programming language was to encourage its adoption by young programmers through teaching of the language in colleges and universities.

Schmidt at RT 1476:9-14
Schwartz at RT 1958:5-20

25. Because the Java programming language lacks basic functionality unless the API packages and associated libraries are present, Sun actively promoted teaching and use of the Java API packages as part of courses in the language.

Astrachan at RT 2091:3-15; 2093:14-17

26. Because Sun actively promoted teaching and use of the Java API packages with the Java programming language, developers who are learning to program in the Java programming language learn how to use the Java API packages, including their SSO, when they learn the language itself.

Bloch at RT 762:13-23
Swetland at RT 961:13-962:3
Morrill at RT 1018:4-23
Bornstein at RT 1769:11-17

27. Because developers are taught the Java API packages together with the language, developers expect and depend on the presence of the Java API packages, including their SSO, to enable them to write programs in the Java programming language.

Astrachan at RT 2202:6-11; 2203:11-15

Q. Professor, do you have an opinion regarding whether or not having these 37 packages in Android is or is not something that’s required to meet the expectations of programmers who are writing in the Java language?

A. I think it’s required to meet expectations of Java programmers.

Q. Professor, do you have an opinion regarding whether having these 37 packages in Android is or is not something that is required to meet industry expectations?

A. I do. I think they are required to meet industry expectations.

Mitchell at RT 2291:1-8

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Q. How about Java, Dr. Mitchell. Let's talk about writing in the Java language. When a programmer is writing a new program in the Java language, he or she expects to have available APIs that will perform all the functions that are in these 37 packages, isn't that right?

A. I think if you said write something in Java, that might be the default assumption, but if you explain more about the context, someone would happily –

Kurian at RT 364:17-21
Screven at RT 519:16-520:6
Mitchell at RT 2289:16-2290:3

28. Because they are a part of the language, programmers often know by heart the fully-qualified method names, which reflect the SSO of the API packages.

Mitchell at RT 2289:24-2290:3

Q. Well, they are just not well-known. Many programmers who have been programming in Java for a long time, they have a lot of these Method signatures memorized; don't they, Professor?

A. Computer geeks are good at memorizing that kind of stuff, yes.
Astrachan at 2169:25-2170:13

29. Books that instruct developers how to write in the Java programming language, including Josh Bloch's award-winning book *Effective Java*, include instruction and discussion about the Java API packages and their SSO.

Bloch at RT 762:24-763:13
Bloch at RT 762:13-23
Swetland at RT 961:13-962:3
Morrill at RT 1018:4-16

30. When companies build their own API packages, they typically rely on and build on top of the standard Java API packages.

Reinhold at RT 685:5-20 (emphasis added)

Q. Do people design their own APIs for Java?

A. Yes, sir. They do all the time.

Q. And when they are doing that -- well, how are they doing that if the APIs are required by the Java programming language?

A. I'm sorry. I don't understand the question. If people are designing their own APIs for Java, then *they are building on top of the Java programming language. They are building typically on top of all of the standard Java APIs* and creating their own APIs for whatever problem they are trying to solve. So, for example, there are -- you know, large financial firms on Wall Street invest heavily in creating their own internally APIs for their own class libraries to handle financial trading activities, but those are not -- those are strictly built on top of all of this Java platform stuff we have been speaking about.

1 31. The functionalities provided by the 37 API packages at issue are all required to
2 make practical use of the Java programming language.

3 Bornstein at RT 1782:6-17

4 Q. Did your determination of what packages would be implemented in
5 the core library have anything to do with what you thought were
6 expectations of Java language programmers?

7 A. Yes, absolutely.

8 Q. And what did it have to do with that?

9 A. Well, you know, I was talking a bit about, like, what's in people's
10 heads as part of what it means to be an API. And so as a Java
11 programmer, as, say, a typical Java programmer, there's certain of
12 these APIs which you just sort of, like, fundamentally think of as
13 kind of part of – part of the system that you can just use without
14 really having to think too much about it.

15 Astrachan 2195:10-2201:17

16 32. Java language programmers and developers have always understood that the Java
17 API packages, along with the Java programming language, were free to use.

18 Swetland at RT 962:4-14

19 Lindholm at RT 861:9-23

20 Bornstein at RT 1769:18-1770:1

21 **5. In order to provide the functionality expected by Java language
22 programmers, the SSO of the Java API packages must be the same.**

23 33. Two platforms are “compatible” if they have common APIs and API packages
24 such that a program written using those APIs and API packages would be understood by and
25 work on both platforms.

26 Mitchell at 2292:25-2293:14

27 Q. So as to the – as to the three lines of code that Dr. Astrachan wrote –
28 where he said give it the address, go get it and read it – you would
expect that code to work on the Java platform, correct?

Q. Yes.

Q. And you would expect it to work on the Android platform, correct,
because it uses the same specifications?

A. If it uses the same API, and the code meets the same specifications
you would expect the same outcome.

Q. So as to those four APIs, as to those four method signatures that
Professor Astrachan used in his program, for those four at least, the
Java platform and the Android platform are compatible, aren't they?

A. Yes, that sounds like a great definition of “compatible” to me.

34. The 37 Java API packages at issue here, including their SSO, are required for
compatibility between implementations of the API packages—including the Android and Java

1 platforms—and to enable preexisting programs written using the methods defined by the API
2 packages to run on new implementations.

3 Bornstein at RT 1787:20-1788:4

4 THE COURT: All right. And as I understand it, you, in fact, wanted to
do that so that the programming community would feel comfortable
5 using the same terminology?

6 THE WITNESS: Yeah. And, actually, not even just a matter of
comfort, but there's a lot of source code out there that wasn't -- you
7 know, wasn't written by -- well, that was written by lots of people
that already existed that could potentially work just fine on Android.
And if we went and changed all the names of things, then that
8 source code wouldn't just work --

9 Mitchell at RT 2286:17-2287:8

10 Q. Now, the question of compatibility. I think this is, needs some
clarification. You heard some back-and-forth, both with Google's
11 counsel and with me, from Dr. Astrachan about when this code
would, quote, run on Android and a Java Platform. Can you explain
to the jury, is Android -- is the Android Platform or the -- let me
12 start over again. Are the Android class libraries compatible with the
Java class libraries?

13 A. I think the point that was illustrated by this code and Dr. Astrachan's
description of it is that, for a given piece of code such as this Class
14 that he wrote with a marker, it may run on both platforms if the only
things it requires are things that are common to the two. And so that
would be generally true about any program. If what it requires is the
15 same on both platforms, it would run on both.

16 Mitchell at RT 2292:25-2293:14

Astrachan at RT 2168:1- 2172:11

Astrachan at RT 2183:2-20

17 35. If a Java programmer does not use the exact name that reflects the SSO of an API
18 package, the programmer's code will not be able to access the prewritten implementing code
19 associated with the relevant method or other element defined in the API package.

20 Bloch at RT 765:3-9

21 Q. And when the programmer speaks to the library to try and get it to
do something for her or him, does the programmer have to speak in
22 very precise language, or general language?

23 A. Very, very precise. If you get anything even a little bit wrong, if you
type a capital letter when the method name has a lower case letter in
Java your program won't run. It won't even compile.

24 Bloch at RT 803:9-20

Lee at RT 1176:17-1177:12

Screven at RT 509:4-16

25 Mitchell at RT 1301:21-1303:7

Mitchell at RT 2291:18-2292:7

26 Astrachan at RT 2154:2-8

27 36. Java and Android are compatible as to the 37 API packages in this case.

28 Astrachan at RT 2171:24-2172:11

1 Mitchell at RT 2292:25-2293:14

2 **B. Equitable Defenses**

3 **1. Sun knew about and approved unlicensed, open source**
4 **implementations of the Java API packages as long as the**
5 **implementation did not use the Java brand.**

6 37. Before Google began developing Android, the GNU project publicly distributed an
7 independent implementation of Sun’s Java platform known as “Classpath.”

8 Schwartz at RT 1972:8-12
9 Bloch at RT 805:10-17

10 38. GNU Classpath used the Java programming language and implemented the
11 specifications of the Java API packages at issue in this case, but GNU did not call its
12 implementation Java.

13 Schwartz at RT 1972:25-1973:3
14 Schwartz at RT 1972:8-12

15 39. Sun was aware of GNU Classpath.

16 Schwartz at RT 1973:4-5

17 40. GNU never took a license from Sun for Classpath.

18 Schwartz at RT 1974:9-12

19 41. Sun never publicly suggested that GNU had done anything wrong by developing
20 and publicly distributing GNU Classpath, much less pursued legal action against GNU.

21 Schwartz at RT 1973:24-1974:8
22 RT at 1863:20-1864:2 (Oracle Response to Google RFA No. 145)

23 42. Beginning in 2005, the Apache Software Foundation (“Apache”) publicly
24 distributed an independent implementation of Sun’s Java SE platform known as “Harmony.”

25 Kurian at RT 397:24-398:4

26 43. Apache Harmony included an independent implementation of the specifications of
27 the Java API packages at issue in this case, including their SSO.

28 Kurian at RT 396:20-397:3

Q. Okay. But Apache had a group of class libraries, correct? You have
to answer verbally, Mr. Kurian.

A. Yes, sir.

Q. And many of those class libraries were in Android. At least that’s
what you told the Google folks?

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

A. I said some of the class libraries were in Android, yes.
Q. Did that include all of the class libraries in the 37 APIs that are accused here in this case?

A. Yes, sir.
Kurian at RT 397:24-398:4

Q. So in 2005, 2006, 2007, 2008, 2009, Apache was using the same libraries and the same APIs we're talking about in this case, without license from Sun, correct?

A. Correct. And Sun had been trying to negotiate with Apache about a license because Apache had petitioned Sun for such a license.

44. Apache never took a license from Sun for Harmony.

Kurian at RT 396:8-9
Screven at RT 561:2-5

45. Sun was aware of Apache Harmony but never publicly suggested that Apache had done anything wrong by developing and publicly distributing Harmony, much less pursued legal action against Apache for Harmony.

Screven at RT 561:25-562:1; 562:17-563:2
RT at 1863:20-1864:2 (Oracle Response to Google RFA No. 145)

46. Sun's public position about Apache Harmony was that as long as Apache did not call its product Java, it could ship Harmony for any purpose, including for use in mobile devices.

TX 2341 (May 9, 2007 article quoting Sun CEO Jonathan Schwartz saying "there is no reason that Apache cannot ship Harmony today. . . .")
Schwartz at RT 2010:5-12

Q. And your testimony is that if they didn't wish to call it Java, this fight was non-existent?

A. I've made that statement time and time again in the media. They are more than happy to ship their -- or we're more than happy for them to ship their code. They just can't call it Java.

Q. Including on mobile devices, sir?
A. Absolutely.

47. The Google executives and engineers responsible for Android were aware of the GNU Classpath and Apache Harmony projects at the time they were developing Android.

Rubin at RT 1688:16-23
Swetland at RT 965:22-966:4
Bornstein at RT 1985:9-1986:1

48. IBM incorporated Apache Harmony code—including implementations of the API packages at issue in this case—into its commercial products.

Schwartz at RT 1977:21-1978:4
Rubin at RT 1689:12-15

1 49. Sun never publicly suggested that IBM had done anything wrong by incorporating
2 Harmony code into its commercial products, much less pursued any legal action against IBM.

3 50. At the time Google was developing Android, Google was aware that IBM was
4 using Apache Harmony code in its commercial products.

5 Rubin at RT 1688:24-1689:15

6 51. During the discussions in the Java Community Process concerning whether Sun
7 should grant Apache a license so that Apache could use the Java trademark, no one from Sun or
8 Oracle ever said that Apache Harmony was infringing copyrights of Sun or Oracle.

9 Lee at RT 1198:16-20
10 Bloch at RT 833:12-17

11 52. Google obtained Harmony code implementing the API packages at issue from
12 Apache, subject to the Apache license, for use in the Android core libraries.

13 Rubin at RT 1684:2-17; 1688:8-23
14 Bornstein at RT 1985:9-1986:1

15 **2. As early as 2005, Sun knew Google intended to implement Java API
16 packages in Android, and Sun never told Google it needed a license to
17 do so.**

18 53. As early as September 19, 2005, Sun knew that Google intended to build a Java-
19 based smartphone, regardless of whether Sun and Google worked together on the project.

20 TX 617 (Andy Rubin to Leo Cizek, September 19, 2005: “If Sun doesn’t
21 want to partner with us to support this initiative, we are fine releasing
22 our work and not calling it Java.”).
23 Rubin at RT 1608:6-10 (discussing TX 617)

24 54. The top management of both Google (Eric Schmidt and Andy Rubin) and Sun
25 (Scott McNealy and Jonathan Schwartz) participated in discussions about a potential partnership
26 between Sun and Google to develop an open-sourced Java-based platform (Android).

27 Schwartz at RT 1978:13-15
28 Schmidt at RT 1486:20-25
 TX 205; TX 435; TX 2372

 55. At no point during the 2005-06 partnership negotiations between Sun and Google
did Sun ever assert to Google that Google needed a license to implement the Java API packages.

 Cizek at RT 1101:7-17

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Q. Mr. Cizek, you described during your testimony earlier that you had a total of three conversations with Google representatives regarding Android; is that right?

A. Yes.

Q. And in each of those conversations there was no discussion, whatsoever, of copyrights that Sun may claim to have, correct?

A. Not in any meeting I was at, correct.

Schmidt at RT 1505:22-1506:2

Q. And was it your understanding that anyone was objecting to Google's use of either the language or the APIs in Android?

A. There was no such objection.

Q. And did anyone at any time ever tell you that you needed a license to use the language or the APIs as part of Android?

A. Uhm, they did not.

56. As a result of the discussions between Sun and Google, Sun knew that Google intended to implement the Java API packages in Android.

Schwartz at RT 1984:21-25

Q. Were you aware that Google was planning to use [the Java API packages] in its product?

A. Yes. We were aware that they were – and I believe they had made statements to the effect that they were creating a Java Linux phone, so they were not subtle about it.

Schmidt at RT 1493:25-1494:3

Q. And are you confident that by this point in time, which is April of '06, Mr. Schwartz was well aware of what the various components of Android would be?

A. Uhm, yes.

57. The partnership negotiations between Sun and Google about the Android platform broke down in mid- to late-2006.

TX 2008 (Cizek contact report entry for May 26, 2006: "After many meetings incl. Alan Brenner, it was agreed that the two companies cannot come to a meeting of minds on how to work together re CDC-HI and open source.").

Cizek at RT 1088:4-22 (discussing TX 2008)

Schmidt at RT 1500:24-1501:1

Rubin at RT 1674:7-8

58. After the negotiations between Google and Sun for an Android partnership broke down, Sun still was aware that Google intended to implement the Java APIs in Android.

Schwartz at RT 1989:2-7

Q. I take it before the time that you wrote this email [TX 3441 (November 9, 2007 email)], you knew that Android would use the Java programming language and a bunch of the Java APIs as well?

A. We knew because everyone in the industry knew.

1 59. In 2007, Sun intentionally elected not to pursue further licensing discussions with
2 Google concerning Android.

3 TX 2009 (March 17, 2007 email from Vineet Gupta to Leo Cizek)
4 Cizek at RT 1091:1-6 (discussing TX 2009)

5 Q. In fact, that's what he told you. He told you to hold off because
6 they're trying to work on bigger stuff with Google; is that right?

7 A. Yes.

8 Q. And you followed Mr. Gupta's instruction, right?

9 A. Yes.

10 60. Sun never told Google it could not release Android without a license to the Java
11 API packages, much less threatened Google with legal action.

12 61. Based on Sun's actions and inactions, Google reasonably believed that it did not
13 need a license from Sun to implement the Java API packages (or their SSO) in Android.

14 Rubin at RT 1691:15-21

15 [Q.] Back in the day when you were developing the Android platform,
16 did you believe you needed a license from Sun to use the Java
17 language APIs?

18 A. Well, no. Specifically we, you know, used some of the APIs that
19 were developed by the Apache Software Foundation and, obviously
20 we, you know, agreed to their Open Source license, but we did not
21 believe that we needed a license from Sun.

22 Schmidt at RT 1505:22-1506:2

23 Q. And was it your understanding that anyone was objecting to
24 Google's use of either the language or the APIs in Android?

25 A. There was no such objection.

26 Q. And did anyone at any time ever tell you that you needed a license
27 to use the language or the APIs as part of Android?

28 A. Uhm, they did not.

 Bornstein at RT 1857:20-1858:6

 Lindholm at RT 861:9-23

3. **After Google publicly announced Android, Sun congratulated Google
and welcomed Google to the Java community.**

62. Google announced the Android platform to the public on November 5, 2007.

 Rubin at RT 1718:22-1719:1

 TX 2352 (November 5, 2007 blog post from Sun CEO Jonathan Schwartz)

63. Jonathan Schwartz, Sun's CEO from 2006-2010, maintained a blog on the Sun
web site that contained official statements of Sun itself.

 Schwartz at 1968:5-15

 Q. Was the blog posted on Sun's website?

 A. The blog was posted on Sun's website and it was our mechanism of
communicating what was important to us, you know, for telling our
shareholders how we were doing, for telling our employees what

1 was important, for telling our customers how to think about our new
products?

2 Q. Did you consider the statements you made on the blog to be official
statements of Sun itself?

3 A. That's exactly what they were. They were the equivalent to me of
4 holding a press conference, but I didn't need to call the press.

5 64. On November 5, 2007, the day Google announced Android, Mr. Schwartz
6 published a post on his official Sun blog congratulating Google on the release of Android and
7 praising Android for "strapp[ing] another set of rockets" onto the Java community's momentum.

8 TX 2352 (November 5, 2007 blog post)
Schwartz at RT 1968:5-15

9 65. As Sun's CEO, Jonathan Schwartz was responsible for all decision-making at Sun
10 during this time, including Sun's licensing and use of its intellectual property.

11 Schwartz at RT 1966:18-1967:3

12 Q. Can you tell the jurors what responsibilities did you have once you
became chief executive officer, Mr. Schwartz?

13 A. Well, I was responsible for all the operations of the company: For
14 setting the vision, for articulating our strategy, for delivering our
performance, for executing on product road maps. As chief
executive, you're responsible for everything that happens in the
company.

15 Q. So would that include licensing and the use of the company's
intellectual property?

16 A. Absolutely. And setting the strategies around our intellectual
property.

17 Ellison at RT 310:21-311:6

18 66. Google's top management (CEO Eric Schmidt and Android project head Andy
19 Rubin) read Mr. Schwartz's blog post at the time it was published, and understood it to mean that
20 Sun approved of Android and would support the platform.

21 Schmidt at RT 1514:8-9
Rubin at RT 1706:4-15

22 Q. So, Mr. Rubin, you read this back in 2007?

23 A. Yes. It came out, you know, the same day or the next day after,
when we had announced Android for the first time.

24 Q. And was this a topic of discussion also at Google?

25 A. Yes, very much so.

26 Q. Were you pleased by the announcement?

27 A. By what was written on the blog, sir?

28 Q. Yes.

A. Yes. Obviously, in my many months and years of negotiating with
Sun, this was, you know, full support of what we were doing with
Android and acknowledging that Sun was supportive of it.

1 67. Members of Google’s Android engineering staff read Mr. Schwartz’s blog post at
2 the time it was published and understood it to mean that Sun approved of Android and would
3 commit its engineering resources to support the platform.

4 Morrill at RT 1025:11-1026:2; 1027:25-1028:11

5 68. On November 9, 2007, Sun CEO Jonathan Schwartz sent an email message to
6 Google CEO Eric Schmidt congratulating Google on the release of Android and offering Sun’s
7 support for the upcoming announcement of the Android Software Development Kit (“SDK”).

8 TX 3441 (Email from Schwartz to Schmidt: “Let us know how we can
9 help support your announcement next week – we’re happy to do so.”)
 Schmidt at RT 1510:14-1512:5 (discussing TX 3441)

10 69. Before the time he wrote the November 9, 2007 email [TX 3441], Mr. Schwartz
11 knew that Google would use in Android the Java language and the Java API packages.

12 Schwartz at RT 1989:2-7.

13 70. After Google’s announcement of Android and Sun CEO Jonathan Schwartz’s blog
14 post, Mr. Schwartz continued to make supportive comments in the market about Android.

15 Schwartz at RT 1991:20-1992:1

16 Q. Now, following the announcement and the posting of your blog, did
 you continue to make supportive comments in the market about
 Android?

17 A. Yes, because there would be no point in standing up and saying, you
18 know, “They are doing something wrong. We didn’t think they were
 doing anything wrong.” We didn’t like it, but we weren’t going to
19 stop it by complaining about it.

20 71. On November 12, 2007, one week after the initial announcement of Android,
21 Google released the Android SDK, which included the Java API packages at issue, and their SSO.

22 Schmidt at RT 1509:3-15

23 Q. Would anyone that wanted to know what APIs Android was using at
 that time have been able to find out?

24 A. Yes, absolutely.

25 Q. How would they do that?

26 A. By looking at the developer kit, because the developer kit would
 state the APIs that were available.

27 Q. So there would be some Android APIs included?

28 A. Yes.

 Q. And some Java APIs?

 A. Yes.

 Q. And anyone that wanted to know what was in it, they could look on
 the website; it would all be there?

 A. Absolutely.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Mitchell at RT 1307:15-23

Q. The organization and structure of the 37 APIs in Android was published on a website back in 2007?

A. I didn't check the date. I don't recall, but I believe that's completely possible.

Q. And, certainly, the organization structure in Android is something that anyone that wanted to could determine back as far as 2007, right?

A. Well, whenever this was available on the web. And someone else could have looked at it just the same way I did.

Rubin at RT 1703:16-21

72. As Sun's CEO, Mr. Schwartz made an affirmative decision not to pursue litigation against Google over Android.

Schwartz at RT 2002:5-7

Q. Mr. Schwartz, as CEO of Sun, did you make a decision not to pursue litigation against Google over Android?

A. Yes. We didn't feel we had any grounds.

Schwartz at RT 2004: 9-14

[A.] What we knew was the Open Source Community was free to create products, just as Oracle and IBM had created Linux. You know, Google and others would go off and create, you know, different handsets. They were independent implementations that may have used portions of our ideas, but so long as they didn't use our code proper, they did nothing wrong.

4. After Google's announcement of Android and release of the SDK, Sun continued to talk with Google and publicly support Android.

73. In the spring of 2008, Sun CEO Jonathan Schwartz met personally with Google CEO Eric Schmidt at Sun's headquarters to discuss opportunities for Sun to get involved with Android.

Schmidt at RT 1521:3-18
TX 3466 (March 31, 2008 Email from Schmidt to Schwartz)
Schwartz at RT 1993:13-21; 1995:9-16

74. Mr. Schwartz did not suggest to Mr. Schmidt at the spring 2008 meeting, or at any other time, that Google had done anything wrong through its implementation of the 37 Java API packages in Android (including its use of the SSO of those Java API packages).

Schmidt at RT 1526:25-1527:15; RT 1528:6-23; 1568:5-14
Schwartz at RT 1996:5-16

Q. Do you remember asking him about the Android licensing program in the meeting?

A. I do. And, again, this was -- we had heard our customers saying they were very suspicious of Google and they were very suspicious of the licensing agreements that had been used to deliver the handset platform. What Google had done is built basically an open source

1 phone and they were telling everybody the world over, “Hey, just
2 use our technology and then you can build a phone.” So what we
3 were trying to tell Google, “That’s fine, but people are suspicious of
4 picking up your technology. If you come to us, we can make them
5 less suspicious.”

Schwartz at RT 1992:2-12

6 Q. And did you actually give interviews in which you said you thought
7 Android was helping Java?

8 A. I did. And to understand that imagine for a moment that Google had
9 selected Microsoft Windows. That was the choice. They could have
10 picked Microsoft Windows or they could pick an Open Source Java
11 implementation. If you were in our shoes, which would you prefer?
12 At least if they picked an Open Source Java implementation, they
13 could be a part of the community. If they had picked something that
14 was completely variant, it would have had no utility to us
15 whatsoever.

16 75. Based on Sun’s public reaction to the Android announcements, and Mr. Schmidt’s
17 meetings with Mr. Schwartz, Google reasonably believed that Sun approved of Android.

Schmidt at RT 1528:13-1529:1

18 Q. Can you tell the jury what you understood?

19 A. My understanding was that what we were doing was permissible.

20 Q. And can you elaborate a little bit more on that? Why did you feel
21 that way?

22 A. Well, because of the sum of my experiences and interactions, the
23 briefings that I have had, I was very comfortable that what we were
24 doing was both legally correct, permitted by the necessary licenses,
25 or lack of licenses, and consistent with the policies of Sun at the
26 time, as well as obviously Google’s.

27 Q. And did Mr. Schwartz ever say anything that contradicted that?

28 A. He did not.

Bornstein at RT 1824:24-1825:7

76. A few months after Google announced Android, Sun (Vineet Gupta) met with
Google (Andy Rubin) to offer Sun’s congratulations on and support for Android.

Rubin at RT 1708:6-14; 1708:23-1709:12

77. While congratulating Google on Android Mr. Gupta also told Mr. Rubin that Sun
had designed a product (named “Flex” or “FX”) to run on top of the Android platform.

Rubin at RT 1708:6-14; 1709:21-1710:2

78. In May 2008, at Sun’s annual Java One conference for Java developers and
partners, Sun publicly demonstrated its JavaFX product running on Android.

TX 3103 (Video of Sun representatives demonstrating Java FX application
running on top of Android at 2008 JavaOne developer conference)
Schwartz at RT 1996:17-20; 1997:19-25 [re TX 3103]

1 79. Mr. Rubin was aware that Sun demonstrated a Java FX application running on top
2 of Android at the 2008 JavaOne developer conference.

3 Rubin at RT 1710:17-1711:11; 1713:2-15

4 80. Sun's public position at the time of the May 2008 JavaOne developer conference
5 was to praise Android for its positive effect on the Java community.

6 TX 2259 at 10 ("Q2 What is Sun's position on Google Android? Sun is
7 pleased to have Google amplify the global momentum behind Java
8 technology the world's most prolific open source software platform on
9 more than six billion devices. We are excited by the Open Handset
Alliance's upcoming open source contributions of new services and
frameworks. We welcome Google to the community and look forward
to collaborating on the evolution of the Java platform as part of our
ongoing relationship.")

10 81. Sun and Google were business partners before Google announced Android, and
11 continued to work together after Google announced Android.

12 TX 2019 (email from Leo Cizek to Jacob Lehrbaum discussing potential
13 revenue opportunities from working with Google around Java FX
Mobile)

14 Schwartz at RT 1986:18-1987:7

15 Cizek at RT 1094:5-25; 1096:4-1097:4

16 Schmidt at RT 1516:9-18

17 **5. Google was aware of and relied on Sun's public statements of approval
18 and acts of support for the Android platform.**

19 82. Andy Rubin, the Google executive in charge of Android, relied on Sun's inaction
20 and affirmative statements of approval by hiring engineers, creating Google applications for
21 Android, and investing time and money to help Google and its partners bring phones to market.

22 Rubin at RT 1715:2-1717:25

23 Q. Yes. Taking all these things together -- the statement by Mr.
24 Schwartz, your meetings with Mr. Gupta, your various interaction
25 with Sun employees, the demonstration at JavaOne -- did you rely
26 on all these things in making decisions about Android going
27 forward?

28 ***

THE WITNESS: Okay. Yes, you know, based on the positive support I
got from, you know, Sun's media posts as well as the meetings that
we had subsequently to our announcement, I did, in fact, uhm, you
know, come to understand that this general support was going to be
good for my business.

Q. And in reliance on that, what steps, if any, did you take vis-a-vis
Android?

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

THE WITNESS: I continued to invest in what our strategy was from the beginning, which was Java as a programming language and a clean-room implementation of these core libraries.

BY MR. VAN NEST:

Q. And by continuing investments, what do you mean?

A. I continued to hire people. We grew. We were still growing. And I continued to hire engineers to work on the effort. I continued to have the team that had been dedicated to the effort, you know, kind of double down and continue to support Java, the programming language and the libraries.

I also invested extensively in building Google applications in the Java programming language. In this time frame, things like Google Maps, the application that allows you to, you know, get mapping information didn't exist. So I had a whole team focused on, essentially, that top box on that architectural diagram, developing their applications in the Java programming language.

Q. Did you also spend time and money working with your handset partners?

A. Uhm, yes. This was leading up to the release of the first Android handset, the HTC handset. These partnerships were complex and very broad. It involved relationships with every handset manufacturer that was, you know, probably in the top ten or so, whether it's a Motorola or a Samsung or an LG or an HTC, Sony Ericsson; all the handset manufacturers that were potential customers of Android or customers of open source we were working with.

Q. And throughout this period, I take it, Mr. Rubin, you were still the head of the Android team?

A. Yes, I was.

Q. And responsible for development of the platform?

A. Yes.

Q. And for whatever investments Google made in Android?

A. The product, the strategy, the business and the engineering.

83. Google further relied on Sun's inaction and affirmative statements of approval for Android by "doubl[ing] down" on Java and declining to investigate any alternatives to the Java programming language.

Rubin at RT 1716:14-19

Rubin at RT 1717:12-17

Q. And did you at that time -- whoops -- do any investigation yourself of whether or not it would be feasible to replace Java language in Android?

A. Not at that time. Obviously, based on the positive press and the positive tone of Sun's interactions, I didn't think that was necessary.

84. In October 2008, during the time period of Sun's inaction and affirmative statements of approval for Android, Google publicly released the full source code for the Android platform, including the implementation of the Java API packages at issue and their SSO.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Rubin at RT 1719:10-14

85. During the time period of Sun’s inaction and affirmative statements of approval for Android, Google increased the number of engineers on the Android team from 5 to 90.

Rubin at RT 1687:10-19

86. During the time period of Sun’s inaction and affirmative statements of approval for Android, Google and its partners brought several smartphones to market, and continue to do so today.

Rubin at RT 1718:1-14; 1719:15-1720:12

87. Even after the public release of the Android source code in October 2008 and launch of several Android-based phones, Sun never suggested to Google or in any public forum that Google did not have the right to use in Android the Java API packages at issue in this case.

6. Oracle initially encouraged Android and tried to partner with Google.

88. In June 2009, after Oracle Corporation announced it was acquiring Sun, Oracle CEO Larry Ellison appeared onstage at the 2009 Java One conference with Sun’s Chairman Scott McNealy, and announced that Oracle planned to keep the Java ecosystem open, was “flattered” by Google’s use of Java in Android, and expected to see many more Android products in the future from “our friends at Google.”

TX 2939 (video of Larry Ellison at JavaOne)
TX 2041 at 10-11 (transcript of Larry Ellison at JavaOne)

89. In early and mid-2010, Mr. Ellison met with Eric Schmidt and Larry Page about potential partnerships between Google and Oracle related to Android, and he tried to sell Oracle’s Java virtual machine to Google for use in Android in place of Google’s Dalvik virtual machine.

TX3450 (Ellison Depo. at 83:16 -25 & 90:23-91:01)
Ellison at RT 340:8-341:13

90. In his meetings with Mr. Schmidt and Mr. Page, Mr. Ellison never suggested that Google needed a license to use the Java API packages or their SSO.

91. In June 2010, Oracle President Safra Catz met with Google to discuss Google’s allegedly improper use of Java in Android, but Ms. Catz did not suggest to Google that Google needed a license to implement the Java API packages or their SSO.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

TX 1074 (June 28, 2010 email from Eustace to Catz, stating: “I have discussed your proposals with Google engineers, lawyers, founders, and executives, and they are not acceptable. We will not pay for code that we are not using, or license IP that we strongly believe we are not violating, and that you refuse to enumerate.”)

Catz at RT 2313:23-2314:7

Q. So to what extent, if at all, did you press Oracle’s position to Mr. Eustace at the meeting in June of 2010?

A. I did. I told him that Android needed to be licensed because of our intellectual property, and that they needed to become compatible.

Q. And did Mr. Eustace have a response?

A. He -- he asked -- he asked specifically, what IP?

92. Prior to filing this lawsuit, neither Sun nor Oracle ever suggested to Google that Google had violated any Sun or Oracle copyrights by implementing the API packages or using the SSO of the Java API packages in Android, or that the SSO of the Java API packages was either protectable or protected under copyright law.

II. CONCLUSIONS OF LAW

A. Copyrightability

1. Section 102(b) of the Copyright Act

1. Oracle bears the burden of proving infringement, and proving its assertion that the structure, sequence and organization of the 37 API packages (“API SSO”) is copyrightable (and thus protectable) is part of that burden of proof.

Jada Toys, Inc. v. Mattel, Inc., 518 F.3d 628, 636 (9th Cir. 2008) (copyright plaintiff “must show that he or she owns the copyright and that defendant copied protected elements of the work”) (quotations marks and citation omitted).

2. The copyright registrations in evidence do not raise a presumption regarding the copyrightability of any elements of the works that are not identified on the face of the registrations. Because the API SSO is not identified on the face of the registrations at issue, Oracle is not entitled to a presumption of copyrightability of the API SSO.

TX 464, 475;
Copyright MSJ Order [Dkt. 433] at 8 (presumptions based on a certificate of registration are not presumptions regarding “*specific elements* of a registered work”)

1 3. Even if Oracle were entitled to a presumption of copyrightability, which it is not,
2 the copyright registrations in evidence do not shift the burden of *persuasion* to Google, and
3 Google has come forward with evidence sufficient to satisfy any burden of *production*.

4 Findings of Fact 1-35

5 Fed. R. Evid 301 (burden shifting applies to burden of production, not
6 burden of persuasion)

7 *Entertainment Res. v. Genesis Creative Group*, 122 F.3d 1211, 1218 (9th
8 Cir. 1997) (where burden of production has been shifted, the defendant
9 “must simply offer some evidence or proof to dispute or deny the
10 plaintiff’s prima facie case of infringement”; if defendant’s evidence
11 raises a “serious question,” this shifts the burden back to plaintiff)

12 4. Section 102(b) of the Copyright Act provides that “[i]n no case does copyright
13 protection for an original work of authorship extend to any idea, procedure, process, system,
14 method of operation, concept, principle, or discovery, regardless of the form in which it is
15 described, explained, illustrated, or embodied in such work.” Thus, ideas, systems and methods
16 of operation are unprotected, even if they are part of “an original work of authorship.” This
17 means, for example, that even if a larger work is copyrightable (e.g., a book about the Java
18 programming language or methods of operating prewritten code libraries described in the book),
19 the copyright for that book does not extend to the ideas, systems or methods of operation that are
20 part of that book (e.g., the Java programming language). Plaintiff must prove that each element it
21 claims was infringed is, by itself, copyrightable.

22 17 U.S.C. § 102(b)

23 *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1446 (9th Cir.

24 1994) (“the party claiming infringement may place ‘no reliance upon
25 any similarity in expression resulting from’ unprotectable elements”
26 (emphasis in original), quoting *Aliotti v. R. Dakin & Co.*, 831 F.2d 898,
27 901 (9th Cir. 1987))

28 5. The API SSO is part of the medium through which Java language developers
express themselves, and is therefore part of an uncopyrightable system or method of operation.

 Findings of Fact 1-3, 8-9, 11-32

Lotus Dev. Corp. v. Borland Int’l, Inc., 49 F.3d 807 (1st Cir. 1995), *aff’d*
 by an equally divided court, 516 U.S. 233 (1996)

Hutchins v. Zoll Med. Corp., 492 F.3d 1377, 1383-84 (Fed. Cir. 2007)

 (“copyright does not protect the technologic process independent of the
program that carries it out”; “Mr. Hutchins’ copyright is limited to
preventing the copying of the specific computer program that he
developed”)

1 *Mitel, Inc. v. Iqtel, Inc.*, 896 F. Supp. 1050, 1055-56 (D. Colo. 1995), *aff'd*
2 124 F.3d 1366 (10th Cir. 1997)

3 6. “Computer programs are, in essence, utilitarian articles—articles that accomplish
4 tasks. As such, they contain many logical, structural, and visual display elements that are dictated
5 by the function to be performed, by considerations of efficiency, or by external factors such as
6 compatibility requirements and industry demands.”

7 *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992)

8 7. Copyright does not protect functional requirements for compatibility.

9 *Sega*, 977 F.2d at 1522 (citing 17 U.S.C. § 102(b))¹
10 *Computer Assoc. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693, 706-10 (2d Cir.
11 1992)

12 *CMM Cable Rep, Inc. v. Ocean Coast Props.*, 97 F.3d 1504, 1519 (1st Cir.
13 1996) (copyright does not protect “‘forms of expression dictated solely
14 at functional considerations’”) (quoting 1 MELVILLE B. NIMMER &
15 DAVID NIMMER, NIMMER ON COPYRIGHT § 2.01[B])

16 *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 46 F.3d 408, 409-
17 10 (5th Cir. 1995), *supplemental opinion clarifying the scope of* 26
18 F.3d 1335 (5th Cir. 1994) (holding that “copyright only protects
19 originality of user interface *to the extent* that the selection of variable
20 inputs from the universe of potential inputs reflects non-functional
21 judgments,” and explaining that “[t]his opinion cannot properly be read
22 . . . to deter achieving compatibility with other models or to the practice
23 employed by users of programs of analyzing application programs to
24 ‘read’ the file formats of other programs”) (citing *Sega*, 977 F.2d at
25 1525-27; *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823,
26 838 (10th Cir. 1993))

27 *Incredible Techs., Inc. v. Virtual Techs., Inc.*, 400 F.3d 1007, 1012 (7th
28 Cir. 2005) (“The exclusion of functional features from copyright
protection grows out of the tension between copyright and patent laws.
Functional features are generally within the domain of the patent
laws.”)

SAS Inst., Inc. v. World Programming Ltd., Case C-406/10 (E.U. Ct.
Justice May 2, 2012),² at ¶¶ 23-24, 29-46 (“[N]either the functionality
of a computer program nor the programming language and the format
of data files used in a computer program in order to exploit certain of
its functions constitute a form of expression of that program and, as
such, are not protected by copyright in computer programs for the
purposes of [Directive 91/250].”)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

¹ Oracle has repeatedly argued that Google mischaracterizes *Sega*. Google’s characterization, however, is precisely correct. See Google 4/5/12 Br. [Dkt. 860] at 5:1-14.

² A copy of this opinion is attached as Exhibit A, and is available at <http://curia.europa.eu/juris/document/document.jsf?text=&docid=122362&pageIndex=0&doclang=EN&mode=req&dir=&occ=first&part=1&cid=147296>.

1 8. The API SSO is functionally required for compatibility with the 37 API packages,
2 and copyright protection therefore does not extend to the structure, sequence and organization of
3 the 37 API packages.

4 Findings of Fact 1-9, 16-22, 33-36
5 Conclusion of Law 7

6 9. Functional requirements for compatibility are unprotected whether the defendant’s
7 work is fully compatible or only partially compatible.

8 *Lotus Dev. Corp. v. Borland Int’l, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995),
9 *aff’d by an equally divided court*, 516 U.S. 233 (1996) (noting that the
10 defendant’s computer program had “many Borland options not
11 available on Lotus 1-2-3”; concluding that the Lotus menu hierarchy is
 an unprotectable method of operation)
 Feist Pubs, Inc. v. Rural Tele. Serv. Co., Inc., 499 U.S. 340, 350 (1991)
 (anyone is allowed “to build freely upon the ideas and information
 conveyed by a work”)

12 10. Regardless of whether the process of designing the API packages required
13 “creativity,” the API SSO is not creative *expression*—it is a *system for* expression, and the form
14 of expression that can be used to express the design ideas is limited and dictated by the Java
15 programming language.

16 Findings of Fact 6, 7, 10
17 Conclusions of Law 5, 8
18 *ATC Distribution Group, Inc. v. Whatever It Takes Transmissions & Parts,*
 Inc., 402 F.3d 700, 707 (6th Cir. 2005) (“Original and creative *ideas*,
 however, are not copyrightable”) (citing 17 U.S.C. § 102(b))

19 11. The API SSO is thus on the unprotectable “idea” side of the idea/expression
20 dichotomy. Section 102(b) of the Copyright Act therefore precludes copyright protection for the
21 API SSO, without regard for its alleged originality, creativity, elegance, “life-changing” nature, or
22 the amount effort it took to develop.

23 17 U.S.C. § 102(b)
24 Conclusions of Law 4-10
25 *Feist*, 499 U.S. at 352-61 (rejecting “sweat of the brow” doctrine)
26 *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 122 (2d Cir. 1930)
 (Learned Hand, J.) (“Though the plaintiff discovered the vein, she
 could not keep it to herself; so defined, the them was too generalized an
 abstraction from what she wrote. It was only part of her ideas.”)
27 *ATC Distribution Group, Inc. v. Whatever It Takes Transmissions & Parts,*
 Inc., 402 F.3d 700, 707 (6th Cir. 2005) (“Original and creative *ideas*,
 however, are not copyrightable”) (citing 17 U.S.C. § 102(b))
28 *Lotus*, 49 F.3d 807

1 *Baker v. Selden*, 101 U.S. 99, 102-04 (1879)
2 *Sega*, 977 F.2d at 1527 (citing *Feist* and rejecting *Sega*'s argument that
3 "considerable time, effort, and money . . . went into development" of its
4 Genesis and Genesis-compatible video games)
5 *SAS Inst., Inc.*, Case C-406/10, at ¶ 40 ("As the Advocate General states in
6 point 57 of his Opinion, to accept that the functionality of a computer
7 program can be protected by copyright would amount to making it
8 possible to monopolise ideas, to the detriment of technological progress
9 and industrial development.")

10 12. The number of elements in the API SSO and their interrelationships, no matter
11 how complex or how long it took to combine those elements, do not render them copyrightable,
12 because section 102(b) provides that "[i]n no case" can ideas, systems or methods of operation be
13 protected by copyright. Sweat of the brow, effort in creating or compiling elements, cannot form
14 the basis for copyright protection.

15 17 U.S.C. § 102(b)
16 *Feist*, 499 U.S. at 352-61

17 2. Merger and *scenes a faire*

18 13. While Google bears the burden of *raising* the issues of merger and *scenes a faire*,
19 Oracle bears the burden of *proof*, because the material in question is not protected if these
20 doctrines apply, and Oracle bears the burden of proving protectability as part of its burden to
21 prove infringement.

22 *Jada Toys*, 518 F.3d at 636 (copyright plaintiff "must show that he or she
23 owns the copyright and that defendant copied protected elements of the
24 work") (quotations marks and citation omitted)
25 *Sega*, 977 F.2d 1524 n.7 (relying on the *lack* of proof of alternatives in
26 concluding that *Sega*'s unlock code was not protected by copyright)
27 *Allen v. Academic Games League of Am., Inc.*, 89 F.3d 614, 618 (9th Cir.
28 1996) (applying merger doctrine to affirm summary judgment based on
plaintiff's failure to prove that "it is possible to distinguish the
expression of the rules of his game manuals from the idea of the rules
themselves")

14. Applicability of the merger and *scenes a faire* doctrines should be assessed based
on the constraints on *Google* at the time of the alleged unauthorized use.

Sega, 977 F.2d at 1524 (copyright protection did not extend to plaintiff's
"unlock" code, which was literally copied, because there was no
evidence that *defendant* had alternatives to using plaintiff's "unlock"
code)
Lotus, 49 F.3d at 818 (to use plaintiff's method of operation, defendant was
required to use "the precise command terms that make up the Lotus
menu command hierarchy")

1 *Baystate Techs. v. Bentley Sys.*, 946 F. Supp. 1079, 1087-90 (D. Mass.
2 1996) (applying *scenes a faire* based on defendant's need to be
compatible with plaintiff's product)

3 15. The merger doctrine is not limited to high levels of abstraction, and indeed can
4 apply even to the literal words of an author's expression for the specific ideas he or she has
5 chosen to express.

6 *Allen*, 89 F.3d 616-18 (applying merger doctrine to prevent plaintiff from
7 asserting copyright protection over his expression of *his particular*
rules for his particular games)

8 16. "In some circumstances, even the exact set of commands used by the programmer
9 is deemed functional rather than creative for purposes of copyright. "[W]hen specific
10 instructions, even though previously copyrighted, are the only and essential means of
11 accomplishing a given task, their later use by another will not amount to infringement."

12 *Sega*, 977 F.2d at 1524 (quoting CONTU Report at 20)
13 *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir.
14 1971) ("When the 'idea' and 'expression' are thus inseparable, copying
the 'expression' will not be barred, since protecting the 'expression' in
15 such circumstances would confer a monopoly on the 'idea' upon the
copyright owner free of the conditions and limitations imposed by the
patent law.")

16 *Baker*, 101 U.S. at 104 (copyright does not protect that which "must
necessarily be used as incident" to an unprotectable idea)

17 *Data East USA, Inc. v. Epyx, Inc.*, 862 F.2d 204, 208 (9th Cir. 1988)
(infringement cannot be based on expression that is "as a practical
18 matter, indispensable or at least standard in the treatment of a given"
idea)

19 *Allen*, 89 F.3d at 617-18 (applying merger doctrine to deny protection to
plaintiff's particular expression of unprotectable game rules)

20 *Computer Assoc.*, 982 F.2d at 707-10

21 *Lexmark Int'l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 535-
36 (6th Cir. 2004) (programming efficiency "figures prominently in the
copyrightability of computer programs")

22 *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532, 1547 (11th Cir. 1996)

23 *Matthew Bender & Co., Inc. v. West Pub. Co.*, 158 F.3d 674, 682 (2d Cir.
1998) (author must make "non-obvious choices from among more than
24 a few options"; copyright does not protect "obvious, garden-variety, or
routine selections")

25 17. At the time Google implemented the API packages at issue, its choices were
26 constrained by the requirements of the Java language, requirements for compatibility with the 37
27 API packages as they existed in J2SE, the expectations and demands of the Java language
28 development community, and widely accepted programming practices within the computer

1 industry generally and specifically among industry users of the Java language. Once these
2 constraints are taken into account, Google was left with no room for creativity, and thus any
3 arguable expression merged with the underlying ideas—which means that copyright cannot
4 protect any arguable expression in the SSO of the API packages.

5 Findings of Fact 1-9, 11-36
6 Conclusions of Law 14-16

7 18. “Under the *scenes a faire* doctrine, when certain commonplace expressions are
8 indispensable and naturally associated with the treatment of a given idea, those expressions are
9 treated like ideas and therefore not protected by copyright.”

10 *Swirsky v. Carey*, 376 F.3d 841, 850 (9th Cir. 2004) (“Under the *scenes a*
11 *faire* doctrine, when certain commonplace expressions are
12 indispensable and naturally associated with the treatment of a given
13 idea, those expressions are treated like ideas and therefore not protected
14 by copyright.”)

15 *Computer Assoc.*, 982 F.2d at 707-10

16 *Gates Rubber*, 9 F.3d at 838 (*scenes a faire* doctrine “excludes from
17 protection those elements of a program that have been dictated by
18 external factors,” including “software standards and compatibility
19 requirements”) (citing *Sega*, 977 F.2d at 1525-27)

20 *Lexmark*, 387 F.3d at 535-36 (programming efficiency “figures
21 prominently in the copyrightability of computer programs”)

22 *Bateman*, 79 F.3d at 1547

23 *Baystate*, 946 F. Supp. at 1087-90 (“Under the *scenes a faire* doctrine,
24 protection is denied to those elements of a program that have been
25 dictated by external factors”)

26 *Mitel*, 896 F. Supp. at 1055-56 (finding command codes to be
27 unprotectable *scenes a faire* where the “proclivities of technicians
28 largely dictate the need to conform the command codes in order to have
market accessibility”)

Matthew Bender & Co., Inc. v. West Pub. Co., 158 F.3d 674, 682 (2d Cir.
1998) (author must make “non-obvious choices from among more than
a few options”; copyright does not protect “obvious, garden-variety, or
routine selections”)

19. At the time Google implemented the APIs at issue, the package, class and method
names defined in the 37 API packages had become commonplace expressions that were
indispensable and naturally associated with the functionality provided by those API packages in
the Java language development community. Google’s freedom of choice was constrained by the
requirements of the Java language, requirements for compatibility with the 37 API packages as
they existed in J2SE, the expectations and demands of the Java language development
community, and widely accepted programming practices within the computer industry generally

1 and specifically within the Java language industry. For these reasons, the API SSO is
2 unprotectable under the *scenes a faire* doctrine.

3 Findings of Fact 1-9, 11-36
4 Conclusions of Law 14, 18

5 20. For these reasons, any arguable expression in the SSO of the API packages has
6 merged into the underlying ideas, or in the alternative the expression API SSO is unprotectable
7 under the *scenes a faire* doctrine.

8 Conclusions of Law 17, 19

9 **B. Equitable Defenses**

10 **1. Laches**

11 21. To prove laches, Google must show by a preponderance of the evidence that (1)
12 Sun and/or Oracle delayed filing a lawsuit concerning the 37 Java API packages for an
13 unreasonably long and inexcusable period of time; and (2) Google has been or will be prejudiced
14 in a significant way due to Sun and/or Oracle's delay in filing the lawsuit.

15 *Danjaq LLC v. Sony Corp.*, 263 F.3d 942, 951 (9th Cir. 2001) (“To
16 demonstrate laches, the defendant must prove both an unreasonable
17 delay by the plaintiff and prejudice to itself.”) (internal quotation marks
18 omitted)

19 *Collegenet, Inc. v. XAP Corp.*, 483 F. Supp. 2d 1058, 1061 (D. Or. 2007)
(applying preponderance of the evidence)

20 22. The relevant period of delay is the period from when Sun/Oracle knew or should
21 have known of the allegedly infringing conduct until the initiation of the lawsuit.

22 *Danjaq LLC v. Sony Corp.*, 263 F.3d 942, 952 (9th Cir. 2001) (“Generally
23 speaking, the relevant delay is the period from when the plaintiff knew
24 (or should have known) of the allegedly infringing conduct, until the
25 initiation of the lawsuit in which the defendant seeks to counterpose the
26 laches defense.”)

27 23. Economic prejudice exists if Google made significant investments in the allegedly
28 infringing product during the period of unreasonable delay.

29 *Danjaq LLC v. Sony Corp.*, 263 F.3d 942, 956 (9th Cir. 2001)
30 *Haas v. Leo Feist, Inc.*, 234 F. 105, 108 (S.D.N.Y. 1916 (L. Hand, J.)) (“It
31 must be obvious to every one familiar with equitable principles that it is
32 inequitable for the owner of a copyright, with full notice of an intended
33 infringement, to stand inactive while the proposed infringer spends
34 large sums of money in its exploitation, and to intervene only when his
35 speculation has proved a success. Delay under such circumstances

1 allows the owner to speculate without risk with the other's money; he
2 cannot possibly lose, and he may win.”)

3 24. Sun and Oracle delayed filing this lawsuit for an unreasonable amount of time.

4 Findings of Fact 32, 37-53, 55-56, 58-60, 62-64, 68-74, 76-77, 88-92

5 25. Google suffered economic prejudice by investing further in the development of
6 Android during the period of Sun's unreasonable delay in initiating this lawsuit.

7 Findings of Fact 52, 61, 66-68, 74-75, 79, 82-86

8 26. For these reasons, Oracle's claim is barred by the affirmative defense of laches.

9 Conclusions of Law 24-25.

10 2. Equitable Estoppel

11 27. To prove equitable estoppel, Google must show by a preponderance of the
12 evidence that (1) Sun and/or Oracle knew of the infringement; (2) Sun and/or Oracle intended that
13 its conduct or communication be acted on, or acted so that Google has a right to believe that Sun
14 and/or Oracle intended that its conduct or communications to be acted on, (3) Google was
15 ignorant of the true facts; and (4) Google relied on Sun and/or Oracle's conduct or
16 communication to Google's injury or material harm.

17 *Hampton v. Paramount Pictures Corp.*, 279 F.2d 100, 104 (9th Cir. 1960)
18 (“Four elements must be present to establish the defense of estoppel:
19 (1) The party to be estopped must know the facts; (2) he must intend
20 that his conduct shall be acted on or must so act that the party asserting
21 the estoppel has a right to believe it is so intended; (3) the latter must be
22 ignorant of the true facts; and (4) he must rely on the former's conduct
23 to his injury.”)

24 *United States v. King Features Ent., Inc.*, 843 F.2d 394 (9th Cir. 1988)
(same).

25 *Hynix Semiconductor Inc. v. Rambus Inc.*, 609 F. Supp. 2d 988, 1025 (N.D.
26 Cal. 2009) aff'd, 645 F.3d 1336 (Fed. Cir. 2011) (applying
27 preponderance of the evidence).

28 28. The conduct in the second prong of the equitable estoppel test can be
accomplished by Sun/Oracle's silence and inaction.

29 *Hampton v. Paramount Pictures Corp.*, 279 F.2d 100, 104 (9th Cir. 1960)
30 (“A holding out may be accomplished by silence and inaction.”)
31 *A.C. Aukerman Co. v. R.L. Chaides Const. Co.*, 960 F.2d 1020, 1042 (Fed.
32 Cir. 1992) (“There is ample subsequent precedent that equitable
estoppel may arise where, coupled with other factors, a patentee's
'misleading conduct' is essentially misleading inaction.”)

1 29. Sun/Oracle's inaction and apparent acquiescence, particularly after its affirmative
2 statements of support, can provide the basis for estopping it from bringing an infringement claim
3 against Google.

4 *Carmichael Lodge No. 2103 v. Leonard*, CIV S-07-2665 LKK/GG, 2009
5 WL 2985476 at *15 (E.D. Cal. Sept. 16, 2009) (noting that "courts that
6 have considered the issue have held that a copyright owner's conduct
7 during a period of permissive use may estopp the owner from later
8 revoking permission and bringing an infringement claim") (citing
9 cases).

10 30. Sun and Oracle knew of Google's use of the Java API packages as early as 2005.
11 Findings of Fact 21-23, 32, 53, 56, 58, 62-64, 69,

12 31. By allowing GNU to distribute its code, publicly endorsing Apache Harmony,
13 posting an official blog approving of Android, congratulating Google's executives privately about
14 Android, demonstrating Sun products on Android devices at public events, and maintaining an
15 ongoing business relationship with Google without ever suggesting to Google that Google's
16 implementation of the Java API packages and use of their SSO infringed Sun's copyrights or that
17 Sun could or would sue Google, Sun acted so that Google had a right to believe that Sun intended
18 its conduct and communication to be acted upon.

19 Findings of Fact 21-23, 32, 37-52, 64-65, 68-70, 72-78, 77-80-81, 87-92

20 32. Google did not know that Sun/Oracle did not intend that its conduct be acted on.

21 Findings of Fact 21-23, 32, 37-52, 61, 66-68, 74-75, 79-82, 91

22 33. Google relied on Sun and/or Oracle's conduct or communication to Google's
23 material harm by investing further in Android development, hiring more Android engineers,
24 further developing the Android code, and entering into agreements with handset partners.

25 Findings of Fact 21-23, 32, 37-50, 52, 61, 66-67, 71, 75-76, 79, 82-86

26 34. For these reasons, Oracle's claim is barred by the affirmative defense of equitable
27 estoppel.

28 Conclusions of Law 30-33.

1 **3. Implied License**

2 35. To prove implied license, Google must show by a preponderance of the evidence
3 that the totality of the parties’ conduct indicates an intent by Sun/Oracle to grant permission to
4 Google to use the SSO of the Java API packages.

5 *Lulirama Ltd., Inc. v. Axxess Broad. Services, Inc.*, 128 F.3d 872, 879 (5th
6 Cir. 1997)
7 *Wang Laboratories, Inc. v. Mitsubishi Electronics America, Inc.*, 103 F.3d
8 1571, 1576, 1581-82 (Fed. Cir. 1997)
9 *Effects Associates, Inc. v. Cohen*, 908 F.2d 55, 558-59 (9th Cir. 1990)
10 *McCoy v. Mitsuboshi Cutlery, Inc.*, 67 F.3d 917, 920 (Fed. Cir. 1995) (“In
11 some circumstances, however, the entire course of conduct between a
12 patent or trademark owner and an accused infringer may create an
13 implied license. *See Stickle v. Heublein, Inc.*, 716 F.2d 1550, 1559, 219
14 USPQ 377, 383 (Fed.Cir.1983). The Supreme Court stated: ‘Any
15 language used by the owner of the patent or any conduct on his part
16 exhibited to another from which that other may properly infer that the
17 owner consents to his use of the patent in making or using it, or selling
18 it, upon which the other acts, constitutes a license and a defense to an
19 action....’ *De Forest Radio Tel. Co. v. United States*, 273 U.S. 236, 241,
20 47 S.Ct. 366, 367, 71 L.Ed. 625 (1927). When warranted by such a
21 course of conduct, the law implies a license.”)

22 36. An implied license may be granted orally, or even implied from the conduct of the
23 party that owns the rights to be licensed.

24 *Effects Associates, Inc. v. Cohen*, 908 F.2d 555, 558 (9th Cir. 1990) (“The
25 leading treatise on copyright law states that ‘[a] nonexclusive license
26 may be granted orally, or may even be implied from conduct.’ Cohen
27 relies on the latter proposition; he insists that, although Effects never
28 gave him a written or oral license, Effects’s conduct created an implied
29 license to use the footage in ‘The Stuff.’”) (quoting 3 M. Nimmer & D.
30 Nimmer, *Nimmer on Copyright* § 10.03[A], at 10-36 (1989))

31 37. The conduct that grants the license may include acts of acquiescence or acts of
32 misrepresentation by Sun and/or Oracle.

33 *Wang Laboratories, Inc. v. Mitsubishi Electronics Am., Inc.*, 103 F.3d
34 1571, 1580 (Fed. Cir. 1997) (“As a result, courts and commentators
35 relate that implied licenses arise by acquiescence, by conduct, by
36 equitable estoppel (estoppel in pais), or by legal estoppel.”)

37 38. The entire course of conduct between Sun and/or Oracle and Google over the
38 relevant time period led Google reasonably to infer consent by Sun and/or Oracle to Google’s
39 making, using, or selling the products that Oracle now claims infringe Oracle’s copyright.

40 Findings of Fact 21-23, 37-50, 52-61, 64-66, 68-72, 73-83, 87-92

1 39. For these reasons, Oracle’s claim is barred by the affirmative defense of implied
2 license.

3 Conclusions of Law 24-25, 29-32, 38

4 **4. Waiver**

5 40. To prove waiver, Google must show by a preponderance of the evidence that
6 Sun/Oracle, with full knowledge of the material facts, intentionally relinquished its rights to
7 enforce the copyrights it asserts.

8 *United States v. King Features Entm’t, Inc.*, 843 F.2d 394, 399 (9th Cir.
9 1988) (“Waiver is the intentional relinquishment of a known right with
knowledge of its existence and the intent to relinquish it.”)

10 41. A waiver may also be implied based on conduct so inconsistent with the intent to
11 enforce a right as to induce a reasonable belief that such right has been relinquished.

12 *Hynix Semiconductor Inc. v. Rambus Inc.*, 645 F.3d 1336, 1348 (Fed. Cir.
13 2011) *cert. denied*, 132 S. Ct. 1540 (U.S. 2012) (“To support a finding
14 of implied waiver in the standard setting organization context, the
15 accused must show by clear and convincing evidence that the
patentee’s conduct was so inconsistent with an intent to enforce its
rights as to induce a reasonable belief that such right has been
16 relinquished.”); *see also Qualcomm Inc. v. Broadcom Corp.*, 548 F.3d
1004, 1020 (Fed. Cir. 2008) (same)

17 42. Sun, with full knowledge of Google’s actions, intentionally relinquished its rights
18 to enforce the copyrights it asserts in the SSO of the 37 Java API packages.

19 Findings of Fact 21-23, 32, 37-53, 56-60, 64, 68-70, 72-74, 76-81, 88-90

20 43. Sun/Oracle’s conduct was so inconsistent with the intent to enforce any rights in
21 the 37 API packages as to induce in Google a reasonable belief that Sun/Oracle had relinquished
22 any rights it may have had in those APIs packages.

23 Findings of Fact 21-23, 32, 37-52, 61, 66-68, 74-75, 82-86, 88-89

24 44. For these reasons, Oracle’s claim is barred by the affirmative defense of waiver.

25 Conclusions of Law 42-43

26 Dated: May 2, 2012

26 KEKER & VAN NEST LLP
27 /s/ Robert A. Van Nest
By: ROBERT A. VAN NEST
Attorneys for Defendant
28 GOOGLE INC.