

Volume 19

Pages 3162 - 3441

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

BEFORE THE HONORABLE WILLIAM H. ALSUP

ORACLE AMERICA, INC.,)	
)	
Plaintiff,)	
)	
VS.)	No. C 10-3561 WHA
)	
GOOGLE, INC.,)	
)	
Defendant.)	San Francisco, California
)	May 9, 2012

TRANSCRIPT OF JURY TRIAL PROCEEDINGS

APPEARANCES:

For Plaintiff:

MORRISON & FOERSTER
755 Page Mill Road
Palo Alto, California 94304

BY: MICHAEL A. JACOBS, ESQUIRE
KENNETH A. KUWAYTI, ESQUIRE
MARC DAVID PETERS, ESQUIRE
DANIEL P. MUINO, ESQUIRE

BOIES, SCHILLER & FLEXNER
333 Main Street
Armonk, New York 10504

BY: DAVID BOIES, ESQUIRE
ALANNA RUTHERFORD, ESQUIRE

(Appearances continued on next page)

Reported By: Katherine Powell Sullivan, RPR, CRR, CSR #5812
Debra L. Pas, RMR, CRR, CSR #11916
Official Reporters - U.S. District Court

Katherine Powell Sullivan, CSR, CRR, RPR
Debra L. Pas, CSR, CRR, RMR
Official Reporters - US District Court - 415-794-6659

APPEARANCES (CONTINUED):

For Plaintiff:

BOIES, SCHILLER & FLEXNER
1999 Harrison Street, Suite 900
Oakland, California 94612

**BY: WILLIAM FRED NORTON, ESQUIRE
STEVEN C. HOLTZMAN, ESQUIRE**

ORACLE AMERICA, INC.
500 Oracle Parkway
Redwood Shores, California 94065

**BY: ANDREW C. TEMKIN, CORPORATE COUNSEL
DORIAN DALEY, GENERAL COUNSEL**

For Defendant:

KEKER & VAN NEST
633 Battery Street
San Francisco, California 94111-1809

**BY: ROBERT ADDY VAN NEST, ESQUIRE
MATTHIAS ANDREAS KAMBER, ESQUIRE
EUGENE MORRIS PAIGE, ESQUIRE
CHRISTA MARTINE ANDERSON, ESQUIRE
MICHAEL S. KWUN, ESQUIRE**

KING & SPALDING LLP
1185 Avenue of the Americas
New York, New York 10036-4003

**BY: BRUCE W. BABER, ESQUIRE
SCOTT T. WEINGAERTNER, ESQUIRE**

GOOGLE, INC.
1600 Amphitheatre Parkway
Mountain View, California 94043

BY: RENNY HWANG, LITIGATION COUNSEL

Also Present:

SAFRA CATZ, President and CFO
Oracle Corporate Representative

CATHERINE LCAVERA
Google Corporate Representative

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

P R O C E E D I N G S

MAY 9, 2012

7:30 A.M.

(The following proceedings were held in open court, outside the presence of the jury.)

THE COURT: Good morning. Please be seated. How is everyone today?

MR. JACOBS: Good.

MR. VAN NEST: Fine. How are you? How are you?

THE COURT: I'm fine. Thank you.

So how can I help you this morning?

MR. JACOBS: Just some housekeeping.

1129 are the demonstratives that were shown during Robert Vandette's testimony.

1130 are the demonstratives shown during Mr. Poore's testimony.

And on a non-housekeeping note, we're going to be calling Mr. Sutphin today.

THE COURT: I have forgotten who he is.

MR. JACOBS: I'm sorry, Mr. Sutphin is our witness to respond to Mr. Schwartz's testimony.

THE COURT: All right.

MR. JACOBS: It will be very focused and short. We do not intend by the testimony we are eliciting to be invading the attorney-client privilege. It will be of a nature that

1 corresponds to the questioning of Mr. Schwartz, which Google
2 asserted in its opposition brief did not amount to a privilege
3 waiver.

4 So we thought ahead to how we're going to approach
5 this. And Google has identified some exhibits they are going
6 to use to cross-examine. And he'll be up, he'll be down, we'll
7 be done with Mr. Sutphin.

8 **THE COURT:** I hear what you're saying. I'm not
9 making any ruling on privilege. I don't know whether what
10 you're saying would or would not waive any privilege. That's
11 for after I hear how it all comes out.

12 **MR. VAN NEST:** Your Honor, I would object to this.
13 I've tried to do everything I can to accommodate counsel.

14 Here's the situation: I agreed to everything they
15 asked for in Phase Two. I said I won't make any reference to
16 Mr. Schwartz' decision. I won't make any reference to the
17 grounds. I won't make any reference to that testimony. I
18 won't call Mr. Schwartz. I'm not going to put it in play in
19 Phase Two.

20 The only reason for this, according to them, is they
21 want a complete commitment as to Phase Three, also. And what I
22 said about that was it's not that I'm not willing to commit to
23 Phase Three, but I would at least like to know what Phase Three
24 is going to look like before I make that commitment.

25 And so what they said in response was, well, no,

1 we've got a witness availability issue. And I said, well, I've
2 been accommodating witness availability issues the whole case.

3 We had Mr. McNealy in here. We moved Mr. Bornstein
4 all around. We've done all that. I don't think just because
5 of a witness availability issue, on a question like this, we
6 should be injecting Mr. Sutphin into Phase Two and then
7 possibly requiring me to call Mr. Schwartz back to respond, and
8 all that. We really don't want to do that.

9 I've agreed for everything they've asked for on Phase
10 Two. Given the fact this is likely going to amount to a waiver
11 and I'm going to be asking for documents, and so on, I really
12 think this is something the Court should prohibit in light of
13 the fact that I'm willing to make a commitment and have made a
14 commitment not to raise the Schwartz issue.

15 And this is only being tendered now based on some
16 witness availability issue, not anything that requires relevant
17 testimony in Phase Two. So I would object to having this
18 witness called now. I think we ought to finish out Phase Two.

19 If he's available end of the day tomorrow and they
20 want to call him at the end of all our Phase Two evidence, that
21 would be a different thing. They want to inject him right in
22 the middle of sort of the key evidence in Phase Two.

23 This is highly objectionable. I have done everything
24 I can to accommodate them.

25 **THE COURT:** Well, the fact remains that at your

1 instance, Mr. Schwartz did say that they thought there were no
2 grounds on which to sue. Even if you don't argue it, the jury
3 may remember it -- it was dramatic testimony -- and hold that
4 against the plaintiff.

5 So I see that point by Oracle. On the other hand, I
6 see your point that this could open up privilege.

7 Now, I want to say, I am not blessing Mr. Jacobs'
8 approach ahead of time. If the privilege gets opened, then the
9 privilege gets opened.

10 I cannot give you a blank check on that, Mr. Jacobs.
11 I hear what you're saying that you don't plan to do it, but
12 that's not enough. You might do it anyway, inadvertently, or
13 due to the way the questioning goes.

14 So that's for a future hour. I won't say future day.
15 Later on this morning we'll have to address that.

16 But don't blame the judge if it turns out that this
17 backfires in some way. So I'm not going to -- the answer is,
18 I'm not going to preclude this.

19 I ask this question just out of curiosity. If you do
20 bring back Mr. -- you said something yesterday that I've been
21 thinking about. If you do bring back Mr. Schwartz, are you
22 then going to put in the fact that the 10-Ks called him out in
23 his blog as a corporate-sponsored way to speak to the public?

24 **MR. VAN NEST:** Sure. I mean, that --

25 **THE COURT:** Mr. Jacobs, I'm going to allow that in.

1 So you be aware that if you put your part in, in order to
2 undermine what Mr. Schwartz says about whether or not they had
3 grounds to sue, and Mr. Schwartz comes back, it's fair game for
4 Mr. Schwartz to rebut what you said in reply with Mr. -- who
5 was it, McNealy? Was that it?

6 **MR. VAN NEST:** McNealy.

7 **THE COURT:** McNealy said that he was not a corporate
8 sponsor.

9 Did Mr. McNealy sign those 10-Ks? Do you know?

10 **MR. JACOBS:** I don't know, Your Honor. I think that
11 the chairman typically doesn't. But I'm not positive.

12 **THE COURT:** Somebody ought to look and see.

13 **MR. VAN NEST:** Well --

14 **THE COURT:** If Mr. McNealy signed those 10-Ks that
15 flat out say that Mr. Schwartz's blog was a communication to
16 the public, you have just as much right to put that in as -- it
17 doesn't even matter whether Mr. McNealy, it's just coloration
18 if he did.

19 But you have just as much right to put that in to
20 straighten out what was said on that subject as Mr. Jacobs has
21 to put in what he wants to put in.

22 And I do think both sides might wisely decide to
23 reach a deal on this and not go down this path.

24 But I cannot -- I feel, as the judge, I cannot
25 prevent you, both sides, from slugging it out in *High Noon*

1 style, Gary Cooper style, just the way you have been doing it.
2 And one of you will be carried out of town with bullet holes in
3 them when the trial is over. Maybe both of you.

4 (Laughter)

5 **THE COURT:** And just like the shootout at the O.K.
6 Corral, or whatever analogy you want to use. So if you two
7 want to litigate the case that way, it is relevant enough that
8 I will let you do it.

9 I also can see reasonable lawyers deciding you are
10 going to focus on the technicalities of the claim limitations
11 and get this part over with.

12 Mr. Van Nest, I cannot grant your motion.

13 **MR. VAN NEST:** Your Honor, I just note that I have
14 Trial Exhibit 971 here. I ask that it be moved into evidence
15 on stipulation. It's the 10-K. It was signed by both McNealy
16 and Schwartz. It's on their exhibit list.

17 I don't need to call Mr. Schwartz for the purpose of
18 getting this in. We've been stipulating to these the whole
19 trial. I have it. It's exhibit 971.

20 **THE COURT:** All right. You're moving 971 in. Any
21 objection?

22 **MR. JACOBS:** Let me see it, Bob.

23 **THE COURT:** Now, be careful on this, because the way
24 you've said it is that if he concedes this you will not call
25 Schwartz. I don't know if you meant that or not.

1 **MR. VAN NEST:** That's not quite what I said.

2 **THE COURT:** All right.

3 **MR. VAN NEST:** I don't think I need Mr. Schwartz to
4 get this in. That's what I said.

5 **THE COURT:** The 10-K.

6 **MR. VAN NEST:** That's right. Depends on what
7 Mr. Sutphin says.

8 **THE COURT:** Be clear because Mr. Jacobs may be
9 understanding you to say that if he stipulates to this one
10 coming in, you will not call Mr. Schwartz. And if that's what
11 you intend to say, fine. But let's be clear on it.

12 Is that what you intend to say?

13 **MR. VAN NEST:** No.

14 **THE COURT:** All right. So this is a separate
15 standalone. You're moving in one exhibit.

16 Is there any objection to this?

17 **MR. JACOBS:** Give me a minute, Your Honor.

18 **MR. VAN NEST:** This is an exhibit on their list, Your
19 Honor.

20 **MR. JACOBS:** Your Honor, I object on the following
21 grounds. The purpose for which Mr. Van Nest wishes to use this
22 is to put an official imprimatur on blog postings.

23 The jury will be confused about the requirements
24 governing filings to the SEC. And, ultimately, we may have to
25 ask the Court for a legal instruction on this point.

1 Companies submit materials as part of their 10-Ks,
2 and refer to -- and their regular updates, in order to avoid
3 any complaint that public statements by company officials have
4 not been adequately disclosed to the investing public. That
5 doesn't necessarily put an official imprimatur on
6 Mr. Schwartz's blog postings.

7 We have blog postings, for example, that are -- from
8 Mr. Schwartz, that are April Fools Day postings, descriptions
9 of pranks. And so while some of his blog postings do talk
10 about the company's business, and it was probably advisable, I
11 imagine, for Sun to make this kind of, in an abundance of
12 caution, securities disclosure, that doesn't mean that a blog
13 posting that says "We welcome Android because it will put
14 rockets on Java" represents a formal statement of Sun's
15 position on whether Android is -- passes legal muster.

16 That kind of confusion about securities law and the
17 implications of securities filings and why companies make
18 securities filings is something we can avoid because this --
19 this doesn't need to come into evidence to make the point that
20 Google is trying to make, which is that Mr. Schwartz was, in a
21 blog posting before Android was actually publicly released,
22 welcomed Android to the Java community.

23 So, on those grounds, we would object to it coming
24 in. It's prejudicial and gives rise to jury confusion and
25 ancillary issues.

1 **THE COURT:** May I see the page? Hand it up to me,
2 please, the 10-K that has the relevant language.

3 Is this in the appropriate time frame? Meaning the
4 time frame where the blog in question appeared.

5 Mr. Van Nest?

6 **MR. VAN NEST:** I'm looking at the dates, Your Honor.
7 This was on Oracle's exhibit list. I'm looking for the date.

8 Guys, help me out here. Where does the date appear?

9 **MR. BOIES:** It's a funny thing. It's on the front
10 page.

11 (Counsel confer off the record.)

12 **MR. VAN NEST:** June 30, 2008, Your Honor. So, yes.

13 **THE COURT:** Here is the relevant statement. This is
14 in part 1 of the 10-K, Item 1, Business under General. And
15 then there is a paragraph that says -- that addresses ways in
16 which investors, the investing public, is notified of material
17 events. Quote:

18 "We periodically webcast company
19 announcements, product launch events and
20 executive presentations which can be viewed
21 via our Investor Relations web site.
22 Additionally, we provide notifications of our
23 material news including SEC filings, investor
24 events, press releases, and CEO blogs as part
25 of the Official Investor Communications

1 section of our Investor Relations web site."

2 Closed quote.

3 So that a reasonable jury could find that that
4 undermines Mr. McNealy's testimony to the effect that
5 Mr. Schwartz's blog, as CEO, was not speaking on behalf of the
6 company. It's up to the jury to make that call, not for the
7 judge. But this is -- this is in the ballpark of a reasonable
8 response to that. So the objection is overruled.

9 Now, is this a self-authenticating document?

10 **MR. VAN NEST:** Yes, Your Honor. It was on their
11 exhibit list. It's an admission by a party. It's signed by
12 Mr. McNealy and Mr. Schwartz, on page 96.

13 **THE COURT:** How could it be on their -- how could it
14 be on their exhibit list as a party admission? They're the
15 party. I don't get that part.

16 But, it was on their list?

17 **MR. VAN NEST:** Yes.

18 **THE COURT:** What's the exhibit number?

19 **MR. VAN NEST:** 971.

20 **THE COURT:** All right. 971 is received in evidence.
21 The objections that have been made are overruled. I'm
22 returning this page to Mr. Van Nest.

23 (Trial Exhibit 971 received in evidence.)

24 **THE COURT:** All right. I want to circle back. I'm
25 not ruling out this witness that Mr. Jacobs wants to bring in.

1 Whose name is what, again?

2 **MR. JACOBS:** I'm sorry?

3 **THE COURT:** The name of that witness.

4 **MR. JACOBS:** Sutphin, Your Honor, S-u-t-p-h-i-n.

5 **THE COURT:** All right. And be aware that I cannot
6 say yes or no on whether you're waiving the privilege. I hear
7 what you're saying, that you don't intend to, but that doesn't
8 necessarily control. And with respect to Mr. Schwartz, I'm not
9 ruling him in or out either. That's for a future day.

10 All right. What's the next item I can help you with?

11 **MR. NORTON:** Your Honor, on the defendant's
12 disclosure list for today is Noel Poore. Mr. Poore is the
13 witness who testified yesterday on performance testing.

14 The disclosures for Mr. Poore, it's not clear to us
15 what issues these go to. So I raised with Google counsel this
16 morning that we have a relevance objection to some, not all, of
17 the 15 or so documents they disclosed yesterday afternoon.

18 **THE COURT:** May I ask a practical question?

19 **MR. NORTON:** Yes.

20 **THE COURT:** He has come and gone. Are we going to
21 get to the defense case today?

22 **MR. NORTON:** We may. I think -- we've indicated to
23 Google that there's a likelihood that we will be able to rest
24 our case today, subject to Mr. Bornstein.

25 Mr. Bornstein was not available this week, and we

1 agreed that we could take him out of order.

2 **THE COURT:** So this would be, if we get to the
3 defense case then they plan to call Mr. Poore back.

4 **MR. NORTON:** Yes, Your Honor.

5 **THE COURT:** And these are exhibits that they want to
6 use. All right. Well, what's wrong with that?

7 **MR. NORTON:** Well, there are some documents that
8 don't appear to be relevant to any issue in Phase Two. And
9 rather than take up their time during the examination of the
10 witness, I tried to resolve any relevance issues. But they
11 refused to tell me what issues those documents are relevant to.

12 And I don't want to be in a situation of getting in
13 the way of their examination, but --

14 **THE COURT:** Give me an example of one that's not
15 relevant.

16 **MR. NORTON:** Sure.

17 **MR. VAN NEST:** Your Honor.

18 **THE COURT:** Yes.

19 **MR. VAN NEST:** Not to interrupt, but I'm wondering if
20 we could have some further meet and confer. I doubt we're
21 going to get to Mr. Poore today, given what I know.

22 **THE COURT:** All right. You two meet at the next
23 break and see if you can work that out.

24 **MR. VAN NEST:** We will.

25 **MR. NORTON:** Thank you, Your Honor.

1 **THE COURT:** We will postpone it.

2 Any other ways I can help you?

3 **MR. VAN NEST:** I don't think there is anything else
4 from us, Your Honor.

5 **MR. JACOBS:** None from us, Your Honor.

6 **THE COURT:** All right. Let's see if the jury is
7 ready to go. They probably are. Go ahead and round them up.
8 While Dawn is lining them up at the doorway, I
9 received a memo called, Oracle America's Brief in Response to
10 Court's Questions. It came in last night.

11 And it covers things like whether or not certain --
12 the judge would -- these are proposals. That the judge would
13 decide some issues and not other issues, or maybe all issues.

14 And I'm not saying yes or no to any of it, of course.
15 But in order to make it not just fall by the wayside as moot, I
16 think, Google, you need to respond to this today. This has
17 proposals in it that I think you could reply to today.

18 **MR. VAN NEST:** Your Honor, I thought -- I think this
19 is part of the --

20 **THE COURT:** Did I give you more time on this already?

21 **MR. VAN NEST:** Yeah. I think this is part of the
22 filings where our response is due tomorrow, anyway, at noon.

23 **THE COURT:** All right. I'll give you until tomorrow.

24 **MR. VAN NEST:** And that's what we --

25 **THE COURT:** We ought to discuss it, then, on Friday

1 because if it's -- if these are going to have any effect on
2 Phase Three, we've got to decide it before Phase Three starts.

3 **MR. VAN NEST:** That's right. I understood that we
4 would be responding to that at, I think, noon tomorrow. And
5 then we have Friday to discuss it.

6 **THE COURT:** All right.

7 **MR. VAN NEST:** Whatever you want --

8 **THE COURT:** That will work.

9 **MR. VAN NEST:** Thank you.

10 **THE COURT:** Are we ready to go? Tell them we're
11 ready, John.

12 Is Mr. Rubin out in the hallway? Let's bring him in.

13 Did you finish your exam of Mr. Rubin, or is there
14 more to go?

15 **MR. JACOBS:** I'm almost done. I have a few things to
16 do, and then we'll be playing some deposition testimony of
17 Mr. Rubin. And then we'll call Mr. McFadden.

18 (Jury enters at 7:52 a.m.)

19 **THE COURT:** Welcome back. Please be seated.

20 How is everybody over there today?

21 (Several jurors respond simultaneously.)

22 **THE COURT:** I see some of you have short-sleeved
23 shirts on, so it's not really too cold in here.

24 **JUROR MS. HOSTYNEK:** Not yet.

25 **JUROR MR. RUTHERFORD:** Wait.

1 go ahead.

2 **BY MR. JACOBS:**

3 **Q.** Could you look, please, at Trial Exhibit 20.

4 Is Trial Exhibit 20 an e-mail exchange including
5 yourself, from March of 2006?

6 **A.** It's -- I'm a cc on the e-mail, yes.

7 **MR. JACOBS:** Offer Exhibit 20 into evidence.

8 **MS. ANDERSON:** No objection, Your Honor.

9 **THE COURT:** 20 received.

10 (Trial Exhibit 20 received in evidence.)

11 **BY MR. JACOBS:**

12 **Q.** Mr. Rubin, true or false, as of August of 2009, you were
13 referring to Dalvik as a Java Virtual Machine?

14 **A.** Uhm, we were using it interchangeably to describe our work
15 at the time internally.

16 **Q.** When you say "interchangeably," what do you mean, sir?

17 **A.** We used the term "JVM" to describe our work internally.

18 **Q.** To describe Dalvik?

19 **A.** Yes.

20 **Q.** In fact, you described Dalvik as recently as August 2009,
21 in internal communications, as a Java Virtual Machine, correct?

22 **A.** I think most of the time I used the term JVM.

23 **Q.** Would you please take a look at Trial Exhibit 219. And
24 look down the -- 219 is an e-mail from you to Alan use cities.

25 Do you see that?

1 A. Yes, I do.

2 Q. Alan Eustace was senior to you at Google, correct?

3 A. He was my manager.

4 Q. And the subject line is, "Hiroshi project
5 responsibilities." Do you see that?

6 A. Yes.

7 Q. And Hiroshi was a member of the Android team, and this
8 e-mail was designed to apprise Mr. Eustace --

9 MS. ANDERSON: Objection, Your Honor. This e-mail is
10 not in evidence. He's reading from it.

11 MR. JACOBS: I'm reading from the subject line, Your
12 Honor.

13 THE COURT: What's the Exhibit number?

14 MR. JACOBS: 219.

15 THE COURT: Do you have an objection?

16 MS. ANDERSON: Yes, we do, Your Honor. It's not in
17 evidence. And it was a document that was late disclosed. It
18 appears that counsel is trying to use it for impeachment, but
19 it actually is not given the witness's testimony.

20 MR. JACOBS: I'll tie it up, Your Honor, with one
21 more question.

22 THE COURT: All right. Go ahead.

23 BY MR. JACOBS:

24 Q. If you look down at the last line under "all of
25 engineering, which is"; do you see that?

1 A. Yes.

2 Q. And the last line reads, "Dalvik (Java Virtual Machine and
3 core libraries)." Do you see that?

4 A. I see that.

5 Q. And does that confirm that, in fact, as of August 2009,
6 you were referring to Dalvik as a Java Virtual Machine?

7 A. I referred to Dalvik as a Java Virtual Machine in this
8 e-mail.

9 MR. JACOBS: No further questions, Your Honor.

10 THE COURT: All right. So we now go to
11 cross-examination.

12 MS. ANDERSON: Thank you, Your Honor.

13 CROSS EXAMINATION

14 BY MS. ANDERSON:

15 Q. Good morning, Mr. Rubin.

16 A. Good morning.

17 Q. Get set up here.

18 Just to refresh the jury a bit, you testified in the
19 first phase of our trial. Would you remind the jury, again,
20 what was your general role in regard to negotiations between
21 Google and Sun?

22 A. I led the negotiations. I managed the Android team, drove
23 the business strategy, and managed the majority of the
24 negotiations, including the negotiations with Sun Microsystems.

25 Q. And in the course of all those negotiations you

1 participated in with Sun while you were at Google during the
2 development of Android, did anyone at Sun ever tell you that
3 Google's Android platform violated any Sun patent?

4 **A.** No, they did not.

5 **Q.** All right. And throughout the entire time period from
6 2005, when you started working at Google, until just before
7 July of 2010, did anybody from Sun or Oracle tell you that
8 Google's Android platform violated any Sun or Oracle patent?

9 **A.** Neither Sun nor Oracle told me that it violated any
10 patents.

11 **Q.** And with regard to the actual Android platform, could you
12 remind the jury, again, about your earlier testimony regarding
13 the release of source code for Android, when Android was
14 actually made available on an open source basis.

15 **A.** Sure. We released the SDK, which was kind of the early
16 version, the beta version of the software, in -- I believe it
17 was November 2007. And then with that -- so you could
18 basically use the system on your PC in an experimental way.

19 And then the entire source code for the stack was
20 available in October of 2008, when we launched the platform
21 officially.

22 **Q.** And is it correct, Mr. Rubin, that it was not until July
23 of 2010 that Oracle ever identified any specific patent as
24 being allegedly violated by the Android platform?

25 **A.** That's correct. And, again, it wasn't -- Oracle didn't

1 bring the patents to my attention. It was between our teams.

2 Q. But it's fair to say, it's your understanding that that
3 was the very first time it was ever raised by Oracle insofar as
4 a specific allegation of infringement of any patent?

5 A. Yes, that's my understanding.

6 Q. All right. Is it also your understanding that this
7 lawsuit was filed around August of 2010?

8 A. Uhm, yes, that's my recollection.

9 Q. Let's take a look at Exhibit 155.

10 MS. ANDERSON: If we could get that up, Ben, please.

11 (Document displayed.)

12 BY MS. ANDERSON:

13 Q. Exhibit 155 was an e-mail that was shown to you by
14 Oracle's counsel, that's dated from November of 2006. Do you
15 see that?

16 A. Yes, I do.

17 Q. All right. And you were asked some questions about this
18 e-mail exchange. And in particular, you were asked about your
19 statement where you said, "They still have patents and
20 trademarks." Do you see that?

21 A. Yes, I do.

22 Q. Could you explain to the jury why you said that in this
23 e-mail?

24 A. Well, the background of the e-mail was that teams were
25 discussing a press announcement that Sun made about the open

1 sourcing of their platform, making the source code available to
2 the world under, I believe it was, the GPL license.

3 And the debate was going on about whether when you
4 take that license, whether that license grants you rights to
5 the patents.

6 I think the debate, obviously, is when you give
7 somebody access to the source code and you encourage them to
8 look at the source code, but you don't give them rights to the
9 patents, you know, what's the trick? You know, why are you
10 making the source code available but not offering them the
11 ability to actually build stuff with it?

12 **Q.** And among the source code, the specific source code you're
13 talking about here is source code you understood in this e-mail
14 exchange was being made available on an open source basis by
15 Sun. Is that right?

16 **A.** Yeah. It was basically Sun's implementation, right? It
17 was Sun's work that they were proposing to open source. It
18 hadn't been open sourced yet, but they were proposing to open
19 source their work.

20 **Q.** When you say Sun's implementation, are you talking about
21 Sun's own proprietary implementation of particular things? Is
22 that right?

23 **A.** That's correct. Sun's proprietary version of Java.

24 **Q.** Included among that, did you understand it included a
25 proprietary version of a virtual machine --

1 A. Yes.

2 Q. -- for Java Language programs?

3 A. Yes.

4 Q. Is that right?

5 A. I understood that.

6 Q. All right. Ultimately, Google developed Android, correct?

7 A. Yes.

8 Q. And part of that particular platform is included a
9 particular virtual machine called the Dalvik Virtual Machine;
10 is that right?

11 A. Yes. We built our own virtual machine technology as part
12 of Android.

13 Q. Did you, Google, use Sun's proprietary implementation of a
14 virtual machine in its Android platform?

15 A. No. They are completely different. We -- Sun had their
16 own implementation. And we created our own implementation.

17 Q. Let's talk a little bit about that.

18 You had been asked questions by Oracle's lawyer
19 earlier about a clean-room implementation and whether it
20 protected against claims of patent infringement. Do you
21 generally remember those questions from yesterday?

22 A. Yes, I remember the question.

23 Q. All right. Well, let's put that in a little context.

24 Google had engineers working on developing, among
25 other things, the virtual machine for the Dalvik Virtual

1 Machine that's part of Android, right?

2 **A.** Yes. The engineers were creating a virtual machine that
3 was part of Android. It was called Dalvik.

4 **Q.** Was Google creating its own implementation for a virtual
5 machine?

6 **A.** Absolutely, yes.

7 **Q.** All right. Were the engineers who were working on the
8 Android team under your supervision, who were specifically
9 working on the virtual machine, provided any guidance as to
10 whether they should study other people's patented inventions in
11 trying to come up with Google's own independent implementation
12 for the virtual machine?

13 **A.** Yes. I -- I believe -- you know, we -- you know, as an
14 engineer, you shouldn't study somebody else's inventions when
15 you're trying to come up with your own.

16 **Q.** And was that guidance provided to the engineers working on
17 the Dalvik virtual machine at Google?

18 **A.** Yes. It was part of our general clean-room guidance.

19 **Q.** And I take it at some point in time the virtual machine
20 that's used as part of the Android platform was completed,
21 right?

22 **A.** Uhm, I mean, it's a work in progress. There's version
23 1.0. Then there's version 2, and 3, and 4. But it was
24 launched with the Android platform when it originally launched.

25 **Q.** Okay. And that was back in the 2007-2008 time frame,

1 right?

2 **A.** That's correct.

3 **Q.** All right. You testified earlier that you didn't do a
4 review of Sun's patent portfolio in connection with that
5 virtual machine release, right?

6 **A.** That's correct.

7 **Q.** Okay. Why is that?

8 **A.** There's a number of reasons. First of all, there was --
9 virtual machines weren't new when Sun did Java. Virtual
10 machines existed before Java. List virtual machine. Even some
11 of the interpreters like Basic had virtual machines like
12 QBasic.

13 And this happened way before Java was even conceived.
14 So we're pretty confident that virtual machines were something
15 that, you know, wasn't rare and wasn't invented by Sun.

16 And then I also think -- I mean, I don't know how
17 many patents are out there. Hundreds and millions of patents
18 worldwide. I think it's just not reasonable to go searching
19 through all this paperwork, especially if you're an engineer.
20 You know, you should be a trained lawyer to do that stuff. And
21 it's just a huge volume of material to sift through.

22 And then, finally, you know, these were separate
23 implementations. We didn't want to go and learn anything from
24 these patent filings that would affect our judgment when we
25 implemented our own version.

1 Q. Now, you testified a lot in Phase One about what you knew
2 about Sun's reaction, publicly and privately, to the release of
3 the Android platform, the open sourcing of it, the making
4 available to the public of the source code. You testified in
5 Phase One about what you observed.

6 Do you generally recall that testimony you gave?

7 A. Yeah, over time -- yes, I do.

8 Q. And you remember, generally speaking, you described Sun's
9 congratulations and support of the Android platform?

10 MR. JACOBS: Your Honor, objection.

11 THE COURT: What's the objection?

12 MR. JACOBS: Bench issues.

13 MS. ANDERSON: I'm just orienting the witness to
14 earlier testimony for follow-up question, Your Honor.

15 THE COURT: Ask that question again.

16 MS. ANDERSON: Sure. Maybe I could rephrase it to
17 address any concern, Your Honor.

18 BY MS. ANDERSON:

19 Q. Did Sun's reaction to the release of the Android platform
20 in any way affect your view of whether a review of Sun's patent
21 portfolio was required?

22 MR. JACOBS: Same objection, Your Honor.

23 MS. ANDERSON: Your Honor, they've opened the door
24 and asked whether a review was conducted. And we need to
25 establish --

1 **THE COURT:** That's true. That was there, and the
2 issue of willfulness is part of the issue for indirect
3 infringement. So the objection is overruled.

4 Please answer the question.

5 **THE WITNESS:** Yeah, I mean, as the evolution of this
6 thing proceeded, obviously, we gained more confidence when we
7 didn't hear from Sun or, you know, Oracle that we were
8 violating any patents.

9 And, of course, all these positive statements were
10 made in the press and media. Congratulatory e-mails were
11 exchanged between the companies. So over the period of the
12 development, you know, we felt it just wasn't necessary,
13 anymore, to worry about this stuff.

14 **BY MS. ANDERSON:**

15 **Q.** Thank you.

16 Let's turn now to talk a little bit about some
17 testimony you gave a few minutes ago about the phrase "Java
18 Virtual Machine" or JVM for short, in connection with virtual
19 machines, and the interchangeable -- calling of the virtual
20 machine Dalvik Virtual Machine or JVM.

21 Do you generally remember those questions from
22 earlier?

23 **A.** Yes, I do.

24 **Q.** All right. Why is it that you periodically would refer to
25 the virtual machine as either a Java virtual machine or a JVM?

1 **A.** It was a virtual machine that was implementing the Java
2 specification. It's not a term we use publicly because the
3 term "Java" is trademarked.

4 But internally I think it was accurate to describe it
5 as something that implemented the Java specification.

6 **Q.** And when you talk about the Java specification, are you
7 referring to the Java Language?

8 **A.** Yes, correct.

9 **Q.** All right. And when you uses the phrase "Java VM" or
10 "JVM" or "Java Virtual Machine" are you referring to Sun's
11 proprietary implementation of a virtual machine?

12 **MR. JACOBS:** Objection. Leading, Your Honor.

13 **THE COURT:** Sustained.

14 **BY MS. ANDERSON:**

15 **Q.** When you've used, in the past, the phrase "Java Virtual
16 Machine" or "JVM" were you or were you not referring to Sun's
17 proprietary implementation of a virtual machine?

18 **A.** Typically, not. It was just a generic term.

19 **Q.** All right. Let's take a look at another exhibit.

20 **MS. ANDERSON:** Exhibit 2714, please, Ben.

21 (Document displayed.)

22 **BY MS. ANDERSON:**

23 **Q.** Do you have that up before you, Mr. Rubin?

24 **A.** I do.

25 **Q.** Okay. Great. You were asked some questions about Exhibit

1 2714. And, in particular, you were asked questions about that
2 sort of first truly full paragraph, that starts with, "Had a
3 long discussion with Eric tonight." Do you see that?

4 **A.** Yes, I see that.

5 **MS. ANDERSON:** If we could get what highlighted,
6 please, Ben. Thank you.

7 **BY MS. ANDERSON:**

8 **Q.** All right. Let's orient ourselves a little bit to the
9 time frame. First of all, this is a 2006 e-mail.

10 Was this in the time frame of the initial
11 negotiations between Google and Sun, that you testified about
12 in the first phase?

13 **A.** Yes. This is when we were in discussions with Sun about a
14 co-development partnership to build the next version of Java.

15 **Q.** Okay. And in this time frame, were you negotiating with
16 Sun about, among other things, the idea of entering into this
17 partnership to license, among other things, various
18 technologies to create this new platform together?

19 **A.** Yes. I mean, generally, the discussions were about
20 cooperatively creating a new version of Java. And,
21 specifically, our ask was that it be open source.

22 **Q.** Okay. And let's talk about that.

23 In that first paragraph that's got the indentations
24 next to it, "had a long discussion," there's a reference in the
25 third sentence where it says:

1 "If you and I can define the open source
2 license and include patent protection, then
3 Eric will be 100 % supportive."

4 Do you see that?

5 **A.** Yes, I do.

6 **Q.** What, if any, relationship is there in that sentence to
7 your reference to open source license and patent protection?

8 **A.** Uhm, well, I think that there's varying types of open
9 source licenses. And the ones we were in favor of using were
10 ones that also granted the licensee rights to the patents of
11 the implementation that was being open sourced.

12 **Q.** And why is that important?

13 **A.** Obviously, if you're -- if you're, you know, going to the
14 great degree of making something available open source, that
15 had previously been proprietary as it was in development, you
16 want people to use it and you want people to adopt it.

17 I wanted a lot of people to build cell phones based
18 on this technology.

19 What I didn't want was for us to open source it with
20 great fanfare, have a lot of people adopt it, and then have Sun
21 chase them around for royalties on patents.

22 **Q.** Okay. And when you made this reference to patent
23 protection in this e-mail, were you alluding to any specific
24 concern about any patent infringement allegation that had ever
25 been made against Google?

1 **A.** Well, yes. I think that, historically, Google had been in
2 discussions with Sun in a separate part of the business.

3 I don't know a great amount of details, but Sun had
4 threatened Google before, in different areas of its business,
5 on patents.

6 **Q.** And specifically, though, in regard to this allegation,
7 had Google made -- strike that. Excuse me.

8 With regard to this particular e-mail, Mr. Rubin, had
9 Sun made any allegation against Google that Android would
10 violate any patent?

11 **A.** No. What Eric's concern was that we were describing in
12 this e-mail was about another issue that Sun had raised with
13 Google.

14 **Q.** Thank you.

15 **MS. ANDERSON:** All right. Let's take a look at
16 another Exhibit, 616, please. Do you have that up? Great.

17 (Document displayed.)

18 **BY MS. ANDERSON:**

19 **Q.** Let's take a look at this exchange. This was an e-mail
20 you were shown by Oracle's counsel earlier. Do you generally
21 recall this?

22 **A.** I -- I don't have the full e-mail here.

23 **Q.** Let you orient yourself. Sure.

24 Okay. And just drawing your attention down to that
25 first e-mail on the first page, the one that says, "Vineet,

1 this is not as good as it sounds." Do you see that one?

2 **A.** Yes, I see that.

3 **Q.** All right. You were asked some questions by Oracle's
4 counsel about this e-mail. So let's again put this in context.
5 When was this e-mail exchange?

6 **A.** Uhm, it was in 2006, February 2006.

7 **Q.** And this was an exchange you were having with a Sun
8 representative, Mr. Gupta; is that right?

9 **A.** Yes, one of the sales leads at Sun.

10 **Q.** All right. And with regard to this e-mail exchange, you
11 testified earlier, in response to questions from Oracle's
12 counsel, that this was in the context of discussions over the
13 joint development that was contemplated at the time. Do you
14 generally remember that?

15 **A.** Yes. This is -- the topic of the e-mail was the joint
16 co-development of the next version of Java.

17 **Q.** And in this particular paragraph, you reference "any
18 tricky behavior well." Do you see that? It's that last
19 sentence there. Do you see that reference to "tricky
20 behavior"?

21 **A.** Yes, I do.

22 **Q.** Would you tell the jury what you meant by that back in
23 2006, when you wrote this?

24 **A.** This is -- I alluded to it in my previous answers. This
25 is when you open source something and make it available under

1 an open source license, I mean, the reason you do that is to
2 get people to use the technology that you are creating.

3 I was -- Eric and I were both concerned that there
4 were, you know, tricks and manipulation of those licenses that
5 allowed Sun to have dual licenses, not be truly open in -- in
6 the endeavor. And this e-mail was referencing that -- all
7 these, you know, terms like, well, it's open source but, or
8 it's open source and the TCK is licensed.

9 Things like that are ways in which you actually
10 control the thing after it's open sourced. And our goal was to
11 make it free and available to anybody.

12 Q. Thank you. Now let's take a look at Exhibit 20, which was
13 shown to you earlier, as well.

14 So this is an e-mail that you were shown earlier from
15 Mr. Horowitz to Mr. Hawthorne, copied to yourself. Do you
16 generally remember that?

17 A. Yes, I do.

18 Q. Okay. And the subject line says "Nedim." Do you see
19 that?

20 A. Yes.

21 Q. And below, a reference to a Nedim Fresko as being a
22 candidate for a title at Sun. Do you see that?

23 A. Yes.

24 Q. First of all, do you have a general recollection of the
25 overall subject matter of this e-mail?

1 A. Yes, I do.

2 Q. All right. Could you just tell the jury, generally, what
3 this e-mail was about?

4 A. Sure. Leslie Hawthorne was a recruiter at the time.
5 Helped us hire people when I was building the team. And the
6 topic of this e-mail was a potential candidate for an
7 engineering role on the team.

8 Q. At the time of this e-mail exchange in 2006, were you
9 still in discussions with Sun over the concept of a joint
10 development of an open source platform with Sun?

11 A. Yeah. I would say we were kind of smack in the middle of
12 those discussions.

13 Q. And among the subjects of discussion were you discussing
14 the concept of purchasing rights to Sun's own implementation of
15 a virtual machine?

16 A. Well, the discussion, you know, obviously, ebbed and
17 flowed with the business negotiations. But I think it's
18 accurate to say that part of those discussions in that time
19 frame were about licensing technology to be open sourced as
20 part of that.

21 Q. And among those being the virtual machine, correct?

22 A. Yeah. Sun's contribution to the co-development was
23 basically to contribute their virtual machine.

24 Q. All right. Is it correct that the concept that was being
25 negotiated over that joint develop, that never actually came to

1 fruition; is that right?

2 **A.** That's right. Our discussions never materialized into a
3 business agreement.

4 **Q.** In regard to this time frame and this discussion of
5 Mr. Fresko, did you have any view as to whether or not
6 Mr. Fresko could be helpful had a joint development partnership
7 actually been entered into with Sun?

8 **A.** Sure. I mean, you know, if Sun's -- if Sun's goal was to
9 open source Java, and the co-development was the next version
10 of Java, having people as members of the team who had knowledge
11 of Sun's implementation of Java that would have been open
12 sourced would be valuable.

13 **Q.** Did Google ever actually hire Mr. Fresko?

14 **A.** No.

15 **Q.** Now, let's take a look at exhibit 22, please. In Exhibit
16 22 -- this was an e-mail that you were shown earlier by
17 Oracle's counsel, that attaches a PowerPoint.

18 **MS. ANDERSON:** If we could see the next page where
19 the PowerPoint starts, Ben.

20 (Document displayed.)

21 **BY MS. ANDERSON:**

22 **Q.** Do you generally recall taking a look at this document
23 earlier?

24 **A.** Yes, I do.

25 **Q.** And let's take a look at page 9 of 24, of this PowerPoint.

1 All right. And you see there in the middle, under
2 "dual license" it says "includes patent grants." Do you see
3 that?

4 **A.** Yes, I do.

5 **Q.** Would you explain to the jury, what is that a reference
6 to?

7 **A.** This is a reference to an open source license. It was one
8 that Sun was proposing. And it was called CDDL. And they were
9 also proposing that it would include patent grants so the
10 people that adopted the open source license wouldn't need to be
11 worried about being chased around after they built products
12 based on the source code.

13 **Q.** And did Google ever use Sun's CDDL?

14 **A.** No.

15 **MS. ANDERSON:** No further questions. Thank you.

16 **THE COURT:** Thank you. Any redirect?

17 **MR. JACOBS:** Yes, Your Honor.

18 **REDIRECT EXAMINATION**

19 **BY MR. JACOBS:**

20 **Q.** So you were concerned that an open source license to Java
21 components, including the Java Virtual Machine, would not
22 protect end users from Sun patents; is that correct?

23 **A.** No, that wasn't my concern.

24 **Q.** I thought you said you were concerned about a trick that
25 Sun might play, that it might allow Google to open source Sun's

1 contributions, and then Sun would chase after implementers or
2 adopters of that open source platform with Sun patents.

3 Isn't that what you meant to say, sir?

4 **A.** That's not what I meant to say.

5 **Q.** What did you mean to say?

6 **A.** What I said was that I was worried that in the joint
7 development of both Sun making contributions and Google making
8 contributions, the resulting open source platform that was to
9 be adopted by manufacturers, not consumers, would give Sun the
10 opportunity, if the license wasn't perfect, to go and extract
11 royalties from the manufacturers as a trick.

12 **Q.** So when I said "adopters," you didn't -- I used the word
13 "adopters," and you substituted "manufacturers," but,
14 otherwise, we are in agreement, correct?

15 **A.** You used the word "consumer." And you also didn't talk
16 about the joint development. You just talked about Sun's work.

17 **Q.** So you were concerned that manufacturers that adopted the
18 product of this proposed collaboration would be pursued by Sun
19 for patent violations?

20 **A.** I -- I would -- I was worried that source code without --
21 source code to an implementation without the right to the
22 patented technology would potentially allow Sun to threaten
23 legal action.

24 **Q.** The open source license that Sun ultimately adopted is the
25 GPL; correct, sir?

1 A. Well, the co-development between the companies did not
2 take place.

3 Q. The open source license that Sun ultimately adopted for
4 the Java platform OpenJDK is the GPL; correct, sir?

5 A. For their implementation of what they open sourced, yes.

6 Q. And you didn't mean to imply that by adopting the GPL, Sun
7 was playing a trick on the open source community; did you, sir?

8 A. We didn't know for sure. But I think if GPL provides
9 patent grants, then there was no trick.

10 Q. And Sun has never asserted and Oracle has never asserted
11 its patents against adopters of the GPL version of the Java
12 platform; correct, sir?

13 A. I don't know.

14 Q. You don't know of any such assertion; correct, sir?

15 A. Correct.

16 Q. Now, you said that up until July 20, no specific patents
17 were brought to Google's attention by Sun and Oracle, right?
18 July 20, 2010, correct?

19 MS. ANDERSON: Objection. Misstates the witness's
20 testimony.

21 MR. JACOBS: Actually, let me do it better, anyway.
22 All right.

23 BY MR. JACOBS:

24 Q. Insofar as Android was concerned, your testimony is that
25 until July 20, 2010, neither Sun nor Oracle brought specific

1 patents to Sun's attention, correct? I'm sorry, Google's
2 attention?

3 **MS. ANDERSON:** Same objection.

4 **THE COURT:** The witness can tell us. Is that what
5 you testified to or not?

6 **THE WITNESS:** I didn't use the word "specific
7 patents" in my testimony. The topic of patents didn't come up,
8 or the threat of us violating patents didn't come up.

9 **BY MR. JACOBS:**

10 **Q.** So, in general, it is your testimony that -- that neither
11 Sun nor Oracle raised patent issues relating to Android until
12 July 20, 2010?

13 **A.** Not that I recall, and not to me.

14 **Q.** So you don't recall a meeting with Mr. Kurian, of Oracle,
15 earlier in 2010, in which the subject of patents was raised?

16 **A.** Uhm, can you refresh my memory on the date?

17 **Q.** Not exactly, but there was such a meeting with Mr. Kurian,
18 correct?

19 **A.** Mr. Kurian of Oracle.

20 **Q.** Yes.

21 **A.** Yes.

22 **Q.** And in that meeting the subject of patents and Android's
23 infringement of Oracle's patents was raised; was it not, sir?

24 **A.** I don't believe so. We were -- I had a number of meetings
25 with Mr. Kurian. They ranged from continuing our discussions

1 about developing the next version of Java. And during those
2 discussions we also talked about the same issues, which is, we
3 need to open source the next version of Java with patent
4 protection for the adopters.

5 **Q.** In fact, sir, wasn't there a discussion in general of
6 patents, in which you participated, which led to -- ultimately
7 led to a meeting between the legal teams on July 20, 2010?

8 **A.** Uhm, I participated in a number of meetings with
9 Mr. Kurian. When we couldn't reach agreement on building the
10 next version of Java together, it culminated in legal meetings.

11 **Q.** After that meeting on July 20, 2010, in which the '104 and
12 '520 patents in this action were brought to Google's attention,
13 you made no changes to Android 2.2 or Android 2.3 on account of
14 those patents; correct, sir?

15 **A.** I personally did not make any changes. And I didn't
16 instruct the team to make changes. But I don't know if the
17 legal team worked directly with -- with the team.

18 **Q.** Let's take a look at Trial Exhibit 230, please.

19 (Document displayed.)

20 **MR. JACOBS:** May I, Your Honor?

21 **THE COURT:** Of course, yes.

22 **THE WITNESS:** Thank you.

23 **BY MR. JACOBS:**

24 **Q.** Now, 230 is an e-mail exchange in 2007, between you and
25 members of the Android team, correct?

1 A. Uhm, yes.

2 Q. And there's a discussion in there about why you don't --
3 why you don't want to adopt for Android the GPL. Do you see
4 that?

5 A. Uhm, we were discussing GPL embedded systems, and why it
6 didn't work for us.

7 Q. And you say:

8 "Sun chose the GPL for this exact reason, so
9 that companies would need to come back to
10 them and take a direct license and pay
11 royalties."

12 Do you see that?

13 A. I see that.

14 Q. And that was your understanding of Sun's attempt to have
15 an open source version under the GPL, but also protect its
16 commercial business, correct?

17 A. No, that's not what I said.

18 Q. Now, at the bottom of that e-mail string you say -- that
19 e-mail on August 11, 2007, you say:

20 "PS, we negotiated nine months with Sun and
21 decided to walk away after they threatened to
22 sue us over patent violations."

23 Do you see that?

24 A. I do.

25 Q. And you wrote that in that e-mail on August 11, 2007, to

1 members of your team; correct, sir?

2 **A.** Yes.

3 **Q.** With whom you were in regular and frequent contact,
4 correct?

5 **A.** Yes.

6 **Q.** Now, let me just ask you this again.

7 Is it your testimony that you had no discussion with
8 Sun about whether there were any patents relating to the Sun
9 virtual machine at any time, that is up -- when I say "Sun"
10 now, I mean Sun and not Oracle America or Oracle.

11 **A.** No, that's not what I testified.

12 **Q.** So you do recall some discussions about patents relating
13 to the Sun Virtual Machine with Sun?

14 **A.** We talked about patents all the time. Sun never accused
15 me of violating their patents, is what I testified.

16 **Q.** So you did have discussions with Sun all the time about
17 patents relating to the virtual machine?

18 **A.** I wanted to make sure that open source versions of our
19 co-development were free of claims of intellectual property by
20 Sun.

21 **Q.** True or false, you had discussions with Sun all the time
22 about patents relating to the virtual machine?

23 **A.** I think "all the time" was colorful language by myself.
24 It's hard to talk about patents all the time.

25 (Laughter)

1 Q. You had discussions with Sun about patents relating to the
2 virtual machine?

3 A. Yes.

4 MR. JACOBS: No further questions, Your Honor.

5 THE COURT: Anything more?

6 MS. ANDERSON: Yes, Your Honor. Very briefly.

7 RECROSS EXAMINATION

8 BY MS. ANDERSON:

9 Q. Mr. Rubin, when you said you had discussions with Sun
10 about patents, generally what did you mean by that?

11 A. That, again, we were trying to create an open platform.

12 My proposal to them was a co-development with a
13 specific open source license that granted patents so Sun
14 wouldn't go and try and extract royalties from the downstream
15 licensees.

16 Q. Did any of those discussions with Sun about patents refer
17 to specific allegations that Google's own independent
18 implementation of a virtual machine violated any Sun patent?

19 A. No, they did not.

20 Q. Let's take a look, quickly, at Exhibit 230, which you were
21 just shown by counsel.

22 MS. ANDERSON: Could we get that up, please, Ben.

23 (Document displayed.)

24 BY MS. ANDERSON:

25 Q. All right. And this was an e-mail exchange from August of

1 2007. And you were asked some questions about it, and
2 specifically about the "PS."

3 **MS. ANDERSON:** Could we have the "PS" section
4 highlighted, please, Ben.

5 **BY MS. ANDERSON:**

6 **Q.** All right. And in this PS there's a reference by you to:

7 "Having negotiated nine months with Sun, but
8 decided to walk away after they threatened to
9 sue us over patent violations."

10 The reference to "patent violations," what is that a
11 reference to, Mr. Rubin?

12 **A.** That was a reference not to threats over what I was doing
13 with my implementation of Java. These -- this is what I
14 referred to as the threats Sun was making to the other part of
15 the company. I wasn't involved in those conversations.

16 But, obviously, when you have two companies trying to
17 form a partnership, you know, and they're threatening another
18 side of the company, you can't, you know, let them play good
19 cop and bad cop with you. So I walked away with my
20 negotiations while the other side of the company settled their
21 difference with Sun.

22 **Q.** Okay. When you say threats to another part of the
23 company, can you explain to the jury what you mean by that.
24 Did those threats have anything to do with Android, whatsoever?

25 **A.** No. It was for technology that related to our servers and

1 hosting and data centers.

2 Q. And the part of the company that has to do with servers
3 and hosting and data centers, that is unrelated to Android; is
4 that right?

5 A. Unrelated to Android and not in my management chain.

6 Q. Thank you.

7 And, finally, briefly, you were asked by counsel
8 about some meetings you had with Mr. Kurian from Oracle. Do
9 you generally remember those questions?

10 A. Yes.

11 Q. Okay. And you -- I believe you testified you weren't sure
12 exactly when those meetings occurred.

13 A. Yeah. I'm pretty vague. We met multiple times.

14 Q. Were those meetings close in time to July of 2010?

15 A. They were, I believe.

16 Q. Okay. Can you give an approximation of about how close in
17 time they were to July of 2010?

18 A. A number of meetings over, probably, like a 30- or 45-day
19 period before.

20 MS. ANDERSON: All right. Thank you. No further
21 questions.

22 MR. JACOBS: One, Your Honor, possible follow-up.

23 **FURTHER REDIRECT EXAMINATION**

24 BY MR. JACOBS:

25 Q. True or false, you did feel, during the Sun discussions,

1 that there was a threat that Sun would pursue legal action if
2 you were unable to successfully complete the partnership?

3 **MS. ANDERSON:** Objection. Vague as to time.

4 **THE COURT:** Overruled. Please answer the question.

5 **THE WITNESS:** Yes, in general, in the early days I
6 was -- I was concerned.

7 **MR. JACOBS:** No further questions.

8 **THE COURT:** All right. May the witness now be
9 excused?

10 **MS. ANDERSON:** Can I ask one follow-up question, Your
11 Honor?

12 **THE COURT:** All right. Go ahead.

13 **MS. ANDERSON:** Thank you.

14 **FURTHER RECROSS EXAMINATION**

15 **BY MS. ANDERSON:**

16 **Q.** Mr. Rubin, with regard to the concerns you just testified
17 about, were those concerns in any way related to concerns that
18 Google's implementation of a virtual machine violated any Sun
19 patent?

20 **A.** Uhm, no, I don't believe so.

21 **MS. ANDERSON:** Thank you. No further questions.

22 **THE COURT:** May the witness now be excused?

23 **MR. JACOBS:** Yes, subject to recall, Your Honor.

24 **THE COURT:** All right. Subject to recall. But
25 you're free to go right now, Mr. Rubin. Have a good day.

1 **THE WITNESS:** Thank you.

2 (Witness steps down.)

3 **THE COURT:** Please leave our exhibits here.

4 It's now time for the next witness.

5 **MR. JACOBS:** We will now hear more from Mr. Rubin, by
6 videotaped deposition.

7 **THE COURT:** How long will this be?

8 **MR. JACOBS:** About 20 minutes, Your Honor.

9 **THE COURT:** Okay. So by now everyone on the jury
10 understands how these depositions work. I won't repeat that.

11 Are there exhibits that the jury needs to be aware of
12 that are problematic, or are they called out by their right
13 name in the transcript?

14 **MR. JACOBS:** I'm sorry, Your Honor. I don't believe
15 so.

16 Oh, yes, there are two, Your Honor, and they are
17 admitted. In the deposition, PX305 is actually TX 155. And
18 PX3 is actually TX 230.

19 **THE COURT:** All right. With that clarification, are
20 we ready to play the 20 minutes' worth? Everyone over there
21 ready to go?

22 (Jurors affirm.)

23 **THE COURT:** Perfect. Go ahead and roll the tape.
24
25

1 **WHEREUPON:**

2 **ANDY RUBIN,**

3 called as a witness for the Plaintiff herein, testified via
4 videotaped deposition played in open court in the presence and
5 hearing of the jury.

6 (Time noted: 8:32 a.m. until 8:53 a.m.)

7 **THE COURT:** Is that it?

8 **MR. JACOBS:** These are it, your Honor. For the
9 record, these are 1128.

10 **THE COURT:** I have that that was 25 minutes. Is
11 there a break now?

12 **MR. JACOBS:** There is, your Honor.

13 **THE COURT:** What would that be?

14 (Brief pause.)

15 **THE COURT:** You can give that to me in a moment.

16 Meanwhile, we'll call the next witness. Unless -- can we start
17 the next witness and go another 20, 25 minutes?

18 (Jurors nodding affirmatively.)

19 **THE COURT:** All right. Everyone is indicating yes,
20 so we'll do that.

21 **MR. JACOBS:** Call Andy McFadden, your Honor.

22 **THE COURT:** All right. You're Mr. McFadden?

23 **THE WITNESS:** Hello.

24 **THE COURT:** Welcome. Please raise your right hand.
25

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

ANDY MCFADDEN,

called as a witness for the Plaintiff herein, having been first duly sworn, was examined and testified as follows:

THE WITNESS: I do.

THE CLERK: Okay. Thank you, Mr. McFadden.

THE COURT: Welcome. While the witness is settling in, Mr. Jacobs, would you retrieve the exhibits that are on the witness bench.

MR. JACOBS: Thank you, your Honor.

THE COURT: And, Mr. McFadden, do you see how close I am to this?

THE WITNESS: Yes.

THE COURT: That's how close you have to be, and it will move all around to make it easy for you.

So why don't you say your name?

THE WITNESS: Andy McFadden.

THE COURT: Perfect.

Go ahead, counsel.

DIRECT EXAMINATION

BY MR. JACOBS:

Q. Good morning, Mr. McFadden. I'm Michael Jacobs.

A. Good morning.

Q. You work on Android at Google, correct?

A. I do.

Q. You have been working on the Android team since July,

1 2005, correct?

2 **A.** Yes.

3 **Q.** I'd like to show you a trial exhibit.

4 **MR. JACOBS:** Your Honor, may I?

5 **THE COURT:** You may.

6 **BY MR. JACOBS:**

7 **Q.** This is 294. Trial Exhibit 294 are what are called
8 snippets that you wrote that cover the period July 18, 2005 to
9 May 5, 2008; correct, sir?

10 **A.** Yes.

11 **MR. JACOBS:** Offer 294 into evidence.

12 **MR. KAMBER:** No objection, your Honor.

13 **BY MR. JACOBS:**

14 **Q.** And snippets are a brief description of what you prepare
15 on a weekly basis to describe what you did, correct?

16 **A.** Correct.

17 **Q.** And it also describes what you're going to do in the
18 following week, correct?

19 **A.** Yes.

20 **Q.** And if you look at the July 18th, 2005 entry --

21 **A.** Okay.

22 **Q.** (Continuing) -- it says:

23 "Get Android build system sorted out. Work
24 on new version of Android runtime."

25 Do you see that?

1 A. Yes.

2 MR. JACOBS: Can we highlight that, Mr. Lee, on page
3 29? Last page.

4 (Document highlighted)

5 BY MR. JACOBS:

6 Q. You worked on the Android runtime, correct?

7 A. I did.

8 Q. And that includes Dalvik, correct?

9 A. Yes.

10 Q. I'd like to now show you your resume?

11 MR. JACOBS: May I?

12 THE COURT: Yes.

13 (Whereupon, document was tendered
14 to the witness.)

15 MR. JACOBS: 955.

16 BY MR. JACOBS:

17 Q. This is your resume?

18 A. Yes.

19 MR. JACOBS: Offer 955.

20 MR. KAMBER: No objection, your Honor.

21 THE COURT: Received in evidence, as well as 294.

22 (Trial Exhibits 955 and received
23 in evidence)

24 MR. KAMBER:
25

1 **BY MR. JACOBS:**

2 **Q.** On the first page of your resume, sir, it states here:

3 "The primary developer of the Dalvik VM
4 runtime."

5 Do you see that?

6 **A.** Yes.

7 **Q.** And that's true; correct, sir?

8 **A.** It was true at the time that I prepared this draft of my
9 resume.

10 **Q.** Which was when, sir?

11 **A.** I think there's a date in here. January, 2010. The date
12 I'm referring to is on the very last page. At the bottom of
13 the text it says, "Last updated 2010/1/10.

14 **Q.** And to give the jury a sense of your career, what have you
15 done since then?

16 **A.** Well, I am still part of Android, but I left the team in
17 April of last year. So up until April of last year, I was
18 working on Dalvik.

19 Since then, when I left Dalvik, I went and worked on
20 the -- what we call the core applications team, specifically
21 the calendar app.

22 **Q.** Let me show you 292, which is another set of snippets.

23 **MR. JACOBS:** May I, your Honor?

24 **THE COURT:** Yes.

25

1 (Whereupon, document was tendered
2 to the witness.)

3 **BY MR. JACOBS:**

4 **Q.** And 292 is your -- is a snippet summary for January 1,
5 2007 to August 31, 2007 that you wrote, sir, correct?

6 **A.** No.

7 **Q.** You did not write it?

8 **A.** I did write it. It is not -- these are not snippets.

9 **Q.** Oh, what are these?

10 **A.** This is a draft of a performance self-evaluation.

11 **THE COURT:** Offer 292 into evidence, your Honor.

12 **MR. KAMBER:** No objection.

13 **THE COURT:** Received.

14 (Trial Exhibit 292 received
15 in evidence)

16 **BY MR. JACOBS:**

17 **Q.** You developed a DEX optimizer that rewrites bytecodes as
18 the class files are uncompressed and created a low-overhead
19 inline native call mechanism, correct?

20 **A.** Yes.

21 **Q.** And that's what you reported here in this draft of your
22 review, correct?

23 **A.** Would you --

24 **Q.** If you look on the screen, we're highlighting it for you.

25 **A.** Oh, excellent.

1 Yes, that is correct.

2 **MR. JACOBS:** May I, your Honor?

3 **THE COURT:** Yes.

4 (Whereupon, document was tendered
5 to the witness.)

6 **BY MR. JACOBS:**

7 **Q.** 302 is your performance self-evaluation for 2006; correct,
8 sir?

9 **A.** Yes.

10 **MR. JACOBS:** Offer 302 into evidence.

11 **MR. KAMBER:** No objection.

12 **THE COURT:** Thank you. Received.

13 (Trial Exhibit 302 received
14 in evidence)

15 **BY MR. JACOBS:**

16 **Q.** Do you take pride in the code you write?

17 **A.** I do.

18 **Q.** Is your code well commented?

19 **A.** Yes.

20 **Q.** And, in particular, the source code files associated with
21 the Dalvik Virtual Machine are well commented, correct?

22 **A.** Yes.

23 **Q.** And, in fact, the Dalvik Virtual Machine source codes have
24 extensive descriptions not only of what, but why; correct, sir?

25 **A.** Correct.

1 Q. And that's recorded here in your self-evaluation; is that
2 correct, sir?

3 A. I'm waiting for the highlight.

4 Q. Yes.

5 **MR. JACOBS:** Scroll down, Mr. Lee. The fourth bullet
6 up from the bottom. "I write good code..."

7 (Document highlighted)

8 A. I see it.

9 **BY MR. JACOBS:**

10 Q. And that was accurate, an accurate self-assessment;
11 correct, sir?

12 A. Yes.

13 Q. Does the virtual machine execute DEX bytecode
14 instructions?

15 A. It does.

16 Q. Is that a terminology that you're comfortable with, "DEX
17 bytecode instructions"?

18 A. I would refer to them as Dalvik bytecode instructions.

19 Q. Or DEX code?

20 A. Okay. Sure.

21 Q. So we'll use those two. And they mean the same thing,
22 right?

23 A. Yeah.

24 Q. So we'll use DEX code, and did you say Dalvik bytecode?

25 A. Dalvik bytecode instructions, yeah.

1 **MR. JACOBS:** 737. May I, your Honor?

2 **THE COURT:** You may.

3 (Whereupon, document was tendered
4 to the witness.)

5 **THE COURT:** 737?

6 **MR. JACOBS:** Yes.

7 **MR. JACOBS:** 737 is admitted already, I believe.

8 **MR. KAMBER:** That's correct your Honor.

9 **THE COURT:** Okay. Thank you.

10 (Document displayed)

11 **BY MR. JACOBS:**

12 **Q.** Can you tell the jury, please, what 737 is?

13 **A.** This is a document that describes the instructions formats
14 used by Dalvik bytecode.

15 **Q.** And, in fact, it reports that the -- that in the DEX file
16 there is a table of names of constants, correct, that's known
17 constant pool?

18 **MR. JACOBS:** Can you scroll up, Mr. Lee? A little
19 further. A little further. Sorry, even a little bit up.

20 (Document highlighted)

21 **BY MR. JACOBS:**

22 **Q.** Do you see the paragraph:

23 "Arguments which indicate a literal constant
24 pool index."

25 **A.** I see it.

1 Q. So can you read that paragraph aloud?

2 A. (As read)

3 "Arguments which indicate a literal constant
4 pool index have the form find at x where find
5 indicates which constant pool is being
6 referred to. Each op code that uses such a
7 format explicitly allows only one kind of
8 constant. See the op code reference to
9 figure out the correspondence. The four
10 kinds of constant pool are string
11 (string_pool_index), type (type_pool_index),
12 field (field_pool_index) and meth,
13 (method_pool_index)."

14 Q. So there are at least four kinds of constant pools in the
15 DEX file; correct, sir? A string, type, field and method?

16 A. Yes.

17 Q. And then the document also explains that some Dalvik
18 instructions can contain vtable offsets and field table
19 offsets, correct?

20 A. Yes.

21 Q. And that's in the paragraph right below the one you just
22 read, correct?

23 A. Yes.

24 Q. Could you read that paragraph aloud, please?

25 A. (As read)

1 "Similar to the representation of constant
2 pool indices, there are also suggested
3 (optional) forms that indicate prelinked
4 offsets or indices. These prelinked values
5 include vtab off (vtable offset), field off
6 (field offset) and iface (interface pool
7 index)."

8 **Q.** Instructions containing constant pool indices are
9 different from instructions containing field offsets; correct,
10 sir?

11 **A.** They are different instructions, but they are not
12 fundamentally different.

13 **Q.** Well, let's see. Lets take a look at 735.

14 **MR. JACOBS:** 735 is already in evidence. May I, your
15 Honor?

16 **THE COURT:** You may.

17 (Whereupon, document was tendered
18 to the witness.)

19 **BY MR. JACOBS:**

20 **Q.** This is a document called "Bytecode for the Dalvik VM."
21 Do you see that, sir?

22 **A.** I do.

23 (Document displayed)

24 **Q.** If you turn to Page 6 of 735, it shows the instruction for
25 the iget instruction. Do you see that, sir?

1 A. I do.

2 Q. Can you explain what the iget instruction is?

3 A. That is the instance field get instruction. What that
4 means is there is an object somewhere and you need to get a
5 piece of data out of it. The data is stored in fields. So
6 what this instruction does is it finds the instance of the
7 object and retrieves the data from the specified field.

8 Q. The reference in this instruction format to field at CCCC,
9 that is the field index; correct, sir?

10 A. Yes.

11 Q. Now, there's another instruction at the very bottom of
12 Page 6, crossing over to Page 7, which is the "Invoke Virtual
13 Instruction;" correct, sir?

14 A. Yes.

15 Q. And in this case the meth@CCCC -- that is, the meth@CCCC
16 and the invoke virtual instruction is the method index;
17 correct, sir?

18 A. Yes.

19 Q. Now, your colleague, Dan Bornstein, gave a talk at the
20 2008 Google IO conference entitled "Dalvik VM Internals." And
21 what I'd like to do is play a clip from that talk and ask you a
22 couple questions on it.

23 This is TX 816 and it's been admitted.

24 (Videotape played in open court;

25 not reported.)

1 Q. Do you agree that -- with Mr. Bornstein that, as he said
2 in that clip, when a DEX file arrives on a device, it will have
3 symbolic references to methods and fields?

4 A. Yes.

5 Q. So the method index in the invoke virtual instruction is a
6 symbolic reference; correct, sir?

7 A. No.

8 Q. Well, let's see. Let's take a look at 294 again. I
9 believe you have that in front of you? Yes.

10 MR. KAMBER: 294?

11 MR. JACOBS: Yes.

12 BY MR. JACOBS:

13 Q. Take a look at your June 26, 2006 entry, please.

14 (Witness complied.)

15 Q. Do you see it says there that you wrote large pieces of
16 class loading/linking/initialization?

17 A. I do.

18 Q. And you do that; is that correct, sir?

19 A. Yes.

20 MR. JACOBS: 47.15 is in evidence. May I, your
21 Honor?

22 THE COURT: Yes, you may.

23 (Whereupon, document was tendered
24 to the witness.)

25

1 BY MR. JACOBS:

2 Q. 47.15 is a code file; correct, sir?

3 A. Yes.

4 Q. And, in fact, the name of this particular file is called
5 "Class.c" from the -- and it's from the Gingerbread version of
6 Android; correct, sir?

7 A. I will take your word for it.

8 Q. Actually, if you look right at the top?

9 A. I see that.

10 Q. And it confirms that this is from Gingerbread; correct,
11 sir?

12 A. I suppose.

13 Q. Now, if you look at Page 2 at Line 58 -- or 59 there is a
14 reference to the "VM spec." Do you see that?

15 A. I do.

16 Q. It says:

17 "VM spec (specifically Version 2 5.4.1)."

18 Correct, sir?

19 A. Yes.

20 Q. And that's a reference to the Java Virtual Machine
21 specification second edition; correct, sir?

22 A. Yes.

23 Q. And that's because Dalvik follows at least some of the
24 Java Virtual Machine semantics when Dalvik does class loading,
25 linking and initialization, correct?

1 A. Correct.

2 Q. And that's because -- and so you were following the Java
3 Virtual Machine specification in order to follow the rules that
4 have to be followed for the Java language; correct, sir?

5 A. Correct.

6 Q. Now, this source code file Class.c, this is part of the
7 code for the Dalvik Virtual Machine, correct?

8 A. Yes.

9 Q. And it has the functions of class loading and class
10 linking; true or false?

11 A. True.

12 Q. And, in fact, we can confirm that if we look at 47.15,
13 Page 1, Lines 8 to 9.

14 (Document displayed)

15 Q. Maybe not Lines 8 to 9. 18 to 19, I'm sorry.

16 Do you see it says:

17 "Class loading, including bootstrap

18 ClassLoader, linking, and initialization."

19 A. Yes.

20 Q. So Class.c has those functions, right, of class loading
21 and class linking?

22 A. Yes.

23 Q. The function -- let me get very technical here for a
24 minute, but I'll see if I can carry it out.

25 The function "dvm find Class No Init In Resolve.c"

1 finds a class and loads it and links it, correct?

2 **A.** I believe the function is in Class.c.

3 **Q.** It finds a class and loads it and links it, correct?

4 **A.** Yes.

5 **Q.** In fact, if we turn to Page 25 of 47.15, Lines 1359 to
6 1373, we see a comment:

7 "Find the named class (by descriptor) up at
8 1360. If it's not already loaded, we load it
9 and link it, but don't execute clinit."

10 Do you see that?

11 **A.** I do.

12 **Q.** And that's true; correct, sir?

13 **A.** Yes.

14 **Q.** And the descriptor is the name of the class, correct?

15 **A.** Yes.

16 **Q.** So "dvmFindClassNoInit," loads and links a class using the
17 name of the class, correct?

18 **A.** Yes.

19 **Q.** And the function "dvmLinkClass" in Class.c converts
20 symbolic references into pointers, doesn't it?

21 **A.** It does.

22 **Q.** And, in fact, that's recorded in the comments also, isn't
23 it, sir, at Page 43, Lines 2428 to 2430?

24 **A.** I'm sorry. What page?

25 **Q.** 43, lines 2428 to 2430.

1 A. Yes.

2 Q. And that comment says:

3 "Link (prepare and resolve). Verification is
4 deferred until later. This converts symbolic
5 references into pointers."

6 Correct, sir?

7 A. Yes.

8 Q. And that's an accurate comment; correct, sir?

9 A. It is.

10 Q. So the Dalvik VM converts symbolic references into
11 pointers during the linking process; correct, sir?

12 A. Yes.

13 Q. The Dalvik VM has a dynamic linking process, correct?

14 A. Yes.

15 Q. The dynamic linking process associates symbolic
16 information with concrete objects?

17 MR. KAMBER: Objection. Form. Vague.

18 BY MR. JACOBS:

19 Q. Correct, sir?

20 THE COURT: If the witness understands it, please
21 answer. Otherwise, rephrase the question.

22 Do you understand the question?

23 THE WITNESS: Not fully.

24 THE COURT: All right. Please rephrase the question.

25

1 **MR. JACOBS:** Your Honor, instead of rephrasing, I
2 would like to play a portion of Mr. McFadden's deposition from
3 May 4th, 2011, lines 6622 to 6709.

4 **THE COURT:** Fine, go ahead.

5 (Brief pause.)

6 **MR. JACOBS:** I will instead read it, your Honor.

7 **THE COURT:** Fine. You can do that, too, but read it
8 exactly and do the entire question and answer.

9 **MR. JACOBS:** Sure.

10 **BY MR. JACOBS:**

11 **"QUESTION:** What kind of steps need to be --
12 I guess what's the difference between
13 loading, linking and initialization in your
14 view?

15 **"ANSWER:** Loading is a slightly inaccurate
16 term in this context in the sense that
17 classes are not loaded individually, but
18 rather mapped into memory en masse as part of
19 a single DEX file."

20 "Linking is the -- in Dalvik it's a dynamic
21 linking process where you associate symbolic
22 information with concrete objects.

23 "And then initialization is a Java language
24 concept in which a class is -- I hate to use
25 the word initialize, but it is initialized

1 and prepared for use."

2 Do you stand by that testimony, sir?

3 **A.** I do.

4 **Q.** And so in Dalvik, linking is a dynamic process in which
5 you associate symbolic information with concrete objects,
6 correct?

7 **A.** In this context, yes.

8 **Q.** Now, that symbolic information is actually represented by
9 indices into the tables in the DEX file, correct?

10 **A.** No.

11 **MR. JACOBS:** Your Honor, I would like to read from
12 Mr. McFadden's deposition, same date, Page 68/14 to 69/7.

13 **THE COURT:** Go ahead.

14 **BY MR. JACOBS:**

15 **"QUESTION:** What steps are involved in the
16 dynamic linking process where you associate
17 symbolic information with concrete objects?

18 **"ANSWER:** The various bits and pieces are
19 identified as indices into tables in the DEX
20 file. Those tables contain pointers to
21 strings that name classes, fields, methods or
22 string constants. The process of linking,
23 for example, a class to its super class would
24 involve getting that index, chasing it
25 through the symbolic information, finding a

1 class -- finding the super class in the form
2 of a class object sitting on the managed
3 heap, and establishing a connection between
4 the two."

5 Do you had stand by that testimony sir?

6 **MR. KAMBER:** Your Honor, that isn't inconsistent with
7 his prior statement. It's not true impeachment.

8 **THE COURT:** Well, nonetheless, it was read into the
9 record. It will stand.

10 Do you stand by that testimony?

11 **THE WITNESS:** I do.

12 **THE COURT:** All right. Go ahead.

13 **BY MR. JACOBS:**

14 **Q.** So the symbolic information is represented by indices into
15 the tables in the DEX file; correct, sir?

16 **A.** I don't agree with that statement.

17 **Q.** The indices lead to strings that name classes, fields,
18 methods or string constants, correct?

19 **A.** Yes.

20 **Q.** During the dynamic linking process, those index -- indexes
21 or indices, I'm using that interchangeably. I apologize.

22 During the dynamic linking process, those indexes are
23 associated with concrete objects in the Dalvik's VM memory,
24 correct?

25 **A.** Yes.

1 Q. The Dalvik VM stores pointers to classes, methods, fields
2 and strings that have been resolved in the past as a
3 performance optimization, correct?

4 A. Yes.

5 Q. Let's take a look at 301.

6 THE COURT: We're going to need to take a break after
7 this next line of questions.

8 MR. JACOBS: May I, your Honor?

9 (Whereupon, document was tendered
10 to the witness.)

11 BY MR. JACOBS:

12 Q. 301 is an email from you dated October 9, 2008, correct?

13 A. Yes.

14 MR. JACOBS: Offer into evidence?

15 MR. KAMBER: No objection.

16 THE COURT: 301 is received.

17 (Trial Exhibit 301 received
18 in evidence)

19 BY MR. JACOBS:

20 Q. And the subject of this email is the "Dalvik Resolved
21 Constant Cache." Do you see that?

22 (Document displayed)

23 A. I do.

24 Q. It begins -- actually, why don't you read what you wrote
25 there, "The Dalvik VM" in the first paragraph?

1 **THE COURT:** Slowly.

2 **A.** (As read)

3 "The Dalvik VM caches pointers to classes,
4 methods, fields and strings that have been
5 resolved in the past as a performance
6 optimization. Unfortunately, this requires a
7 fair bit of storage on the native heap,
8 accounting for about 800k of native heap in
9 the system server."

10 **BY MR. JACOBS:**

11 **Q.** And this is an accurate description of the way the Dalvik
12 VM works; correct, sir?

13 **A.** Yes.

14 **Q.** And by "caching" you mean storing for later use, correct?

15 **A.** Yes.

16 **MR. JACOBS:** Your Honor, this would be a good point.

17 **THE COURT:** All right, 15 minutes. Please remember
18 the admonitions. See you in a few moments.

19 **THE CLERK:** All rise.

20 (Jury exits courtroom at 9:20 a.m.)

21 **THE COURT:** All right. Everyone else be seated.

22 Mr. McFadden, you can take a break, too. Remember,
23 you're on cross-examination, so no talking to the lawyers.

24 Thank you. You can step outside if you wish.

25 (Witness steps outside.)

1 **THE COURT:** All right. Everyone else be seated.

2 Let me ask a question. Was this a 30(b)6 deposition?

3 What kind of deposition was it?

4 **MR. KAMBER:** It was just a fact witness deposition.

5 **THE COURT:** Why isn't it a party deposition?

6 **MR. KAMBER:** It is a party deposition.

7 **THE COURT:** Well, then it can be read for any purpose
8 and it doesn't have to be for impeachment.

9 **MR. KAMBER:** Understood. I believe he said he was
10 reading for impeachment, so.

11 **THE COURT:** If it's a party deposition, you just read
12 it. You don't -- any purpose, subject to Rule 403 only. So
13 you don't have to -- it doesn't have to be impeachment.

14 **MR. KAMBER:** Understood, your Honor.

15 **THE COURT:** So anything else you want to bring up?

16 **MR. JACOBS:** Nothing from us. We're all set, your
17 Honor.

18 **THE COURT:** Thank you. We'll take our 15 minutes.

19 (Whereupon there was a recess in the proceedings
20 from 9:21 a.m. until 9:45 a.m.)

21 **THE COURT:** Welcome again, please be seated.

22 Please go right ahead.

23 **BY MR. JACOBS:**

24 **Q.** Mr. McFadden, I've placed a couple of exhibits in front of
25 you, including Trial Exhibit 46.14, which is a file called

1 "Reduced constants.c." Do you see that?

2 **A.** I do.

3 **Q.** You wrote the file; correct, sir?

4 **A.** Yes.

5 **Q.** And the "Overview" section is something you wrote;
6 correct, sir?

7 **A.** Yes.

8 **MR. JACOBS:** Can we scroll down to "Overview",
9 Line 29?

10 (Document displayed)

11 **BY MR. JACOBS:**

12 **Q.** And the "Overview" is a description of how Dalvik works,
13 correct?

14 **A.** Yes.

15 **Q.** And maybe could you read slowly the first paragraph of
16 what you wrote there on the 46.14?

17 **A.** Okay.

18 "When a class, method, field, or string
19 constant is referred to from Dalvik bytecode,
20 the reference takes the form of an integer
21 index value. This value indexes these into
22 an array of type_id_item, method_id_item,
23 field_id_item or string_id_item in the DEX
24 file. The first three themselves contain
25 (directly or indirectly) indexes to strings

1 that the resolver uses to convert the
2 instruction stream index into a pointer to
3 the appropriate object or struct."

4 **Q.** So when a class, method, field or string constant is
5 referred to from the Dalvik bytecode, the reference takes the
6 form of an integer index value, correct?

7 **A.** Yes.

8 **Q.** And the index value contained in the DEX instruction
9 indexes into an array of type_id_item, method_id_item,
10 field_id_item or string_id_item in the DEX file, correct?

11 **A.** Yes.

12 **Q.** And then the resolver converts the instruction stream
13 index into a pointer to the appropriate object or struct,
14 correct?

15 **A.** Yes.

16 **Q.** Would you agree that in this context the pointer to the
17 appropriate object or struct is a numeric reference?

18 **A.** Yes.

19 **Q.** Would you agree that the instruction stream index is not a
20 numeric reference to the appropriate object?

21 **A.** No.

22 **Q.** Well, the Dalvik VM resolves a Dalvik bytecode reference
23 to a class, method, field or string constant into a pointer to
24 the appropriate object or struct, correct?

25 **A.** Yes.

1 Q. The instruction stream index is not the numeric memory
2 location of the appropriate object or struct, is it?

3 A. No.

4 Q. No, it is not the numeric memory location; correct?

5 A. It is not the address of that item.

6 Q. Because if it were, there would be no reason to convert it
7 into a pointer; true?

8 A. Correct.

9 Q. If it were, it would already be a pointer, true?

10 A. Yes.

11 Q. So the resolver uses the strings in the DEX file to
12 convert the instruction stream index into a pointer, correct?

13 A. Yes.

14 Q. And it turns out that that process is fairly expensive.

15 If you look down at Line 48 to 52 of the file in
16 front of you. That process is fairly expensive; correct, sir?

17 A. Yes.

18 Q. And so -- and so you wrote:

19 "After the first time it completes
20 successfully, the VM records that the method
21 index resolved to a specific method struct.
22 On subsequent execution, the VM just pulls
23 the method pointer out of the
24 resolved-methods array. A similar approach
25 is used with the indexes for classes, fields,

1 and string constants."

2 I read that accurately; correct, sir?

3 A. I believe so.

4 Q. And that description of the way Dalvik works is correct?

5 A. Yes.

6 Q. The Dalvik VM stores pointers that result from resolving
7 the indexes?

8 A. Yes.

9 Q. And the Dalvik VM then pulls them out of storage on
10 subsequent Dalvik bytecode executions?

11 A. Yes.

12 Q. Now, I believe I placed in front of you 47.6 -- actually,
13 I need to ask you one thing just on the previous exhibit.

14 On 46.14, this file, you wrote a note. "This is an
15 incomplete experimental feature," do you see that?

16 A. Yes.

17 Q. But the description we just read, that's an accurate
18 description of Dalvik, correct?

19 A. Yes.

20 Q. Okay. Now let's go to 47.6.

21 Now 47.6 is Resolve.c again for Gingerbread. Do you
22 see that?

23 A. I do.

24 Q. Now, the source code file Resolve.c, that's part of the
25 code for the Dalvik Virtual Machine?

1 A. Yes.

2 Q. And the functions in Resolve.c resolve classes, methods,
3 fields and strings?

4 A. Yes.

5 Q. The functions -- in fact, if you look at the functions in
6 Resolve.c's comment at Line 18, it says it resolves classes
7 methods, fields and strings. Do you see that?

8 A. I do.

9 Q. And that's accurate?

10 A. Yes.

11 Q. You wrote Resolve.c?

12 A. Yes.

13 Q. You started writing Resolve.c in mid to late 2006?

14 A. Sounds about right.

15 Q. And you finished it also in -- around mid to late 2006?

16 A. Substantially.

17 Q. What was unfinished?

18 A. I think we've tweaked it a few times since then, but it
19 was substantially complete at that time.

20 Q. The function dvmResolveClass finds the class corresponding
21 to classIdx which maps to a class name string, correct?

22 A. Yes.

23 Q. And, in fact, that's in the comments at Line 36, where it
24 reads:

25 "Find the class corresponding to 'classIdx'

1 which maps to a class name string."

2 Do you see that?

3 **A.** I do.

4 **Q.** "classIdx," is that how you would read that phrase or
5 would you read it as "class index"?

6 **A.** I have been known to go either way.

7 **Q.** So either will work?

8 **A.** Yes.

9 **Q.** The "class name string," that's the name of the class;
10 correct, sir?

11 **A.** Yes.

12 **Q.** So dvmResolveClass uses the name of the class to locate
13 the corresponding class object and memory; true?

14 **A.** Yes.

15 **Q.** And, in fact, just to avoid any doubt, that's in the code
16 at Line 95 on Page 2, correct?

17 **A.** Yes.

18 **Q.** Now, after the Dalvik Virtual Machine looks up where a
19 class is located in memory, it stores a copy of the lookup in
20 the resolved class table, true?

21 **A.** Yes.

22 **Q.** In fact, we can see that at line 46 back on the previous
23 page, where it says:

24 "We cache a copy of the lookup in the DEX

25 File's resolved class tables so torture

1 references to classIdx are faster."

2 Do you see that?

3 A. I do.

4 Q. And that's accurate; correct, sir?

5 A. Yes.

6 Q. And, again, by "cache" we mean storing for later use,
7 correct?

8 A. Correct.

9 Q. So let's see if we can summarize this. DvmResolveClass
10 takes as an input the class index, or classIdx, locates the
11 corresponding class object in memory and then stores that
12 location so a future reference to classIdx is faster, correct?

13 A. Yes.

14 Q. And in addition to classes, Resolve.c has functions for
15 resolving methods, fields and strings, true?

16 A. True.

17 Q. And Resolve.c resolves methods, fields and strings in
18 approximately the same way that it resolves classes?

19 A. At a high level.

20 Q. At the level we have been discussing?

21 A. Sure.

22 Q. And so at first when you ask it to resolve a class, it
23 checks to see if it's resolved already and stores an entry in
24 the resolved items table, correct?

25 A. Correct.

1 Q. And if there isn't something, if there isn't an entry, it
2 will proceed to attempt to resolve the symbolic reference?

3 A. Correct.

4 Q. And if it succeeds, then it will store the pointer in the
5 resolved items table?

6 A. Yep.

7 Q. And using the resolved items table instead of resolving
8 each time gives performance benefits to the Dalvik Virtual
9 Machine?

10 A. Yes.

11 Q. And the performance is improved by storing the results of
12 resolution because if you didn't store the results, then we
13 would have to repeat the resolver process every time something
14 referred to a class field, a method or a string in the
15 instruction stream; true?

16 A. If it refers to it by the index, then yes.

17 Q. If it referred to a class, field, method or string in the
18 instruction stream, you mean if it referred to the class,
19 field, method or string by an index in the instruction stream;
20 is that how you would correct it?

21 A. By the class index, field index, method index, string
22 index.

23 Q. And Dalvik VM gets performance benefit because after the
24 resolves the names once, it rarely has to resolve the same
25 named twice, correct?

1 A. True.

2 Q. And if it didn't store the results of resolution, Dalvik
3 would have to resolve a symbolic reference every time it
4 encountered it, correct?

5 A. Not necessarily.

6 Q. You get performance benefits by not having to repeat the
7 resolver process, correct?

8 A. Yes.

9 Q. Do you have 739 in front of you? I don't think you do.
10 Let me get you a copy.

11 (Brief pause.)

12 MR. JACOBS: May I, your Honor?

13 THE COURT: You may.

14 BY MR. JACOBS:

15 Q. I'm handing you 739.

16 (Whereupon, document was tendered
17 to the witness.)

18 Q. Did you write this document?

19 A. I did.

20 MR. JACOBS: 739 in evidence.

21 MR. KAMBER: I believe this is in evidence.

22 THE COURT: You think or you know?

23 MR. KAMBER: I know that it's in evidence.

24 THE COURT: All right. Then it's in evidence. We
25 don't have to worry about it.

1 **BY MR. JACOBS:**

2 **Q.** Did you write 739?

3 **A.** I did.

4 **Q.** And this describes Dalvik optimization and verification
5 with dexopt; correct, sir?

6 **A.** It does.

7 **Q.** Now, what we were just talking about a few minutes ago,
8 that was a different area of Dalvik, correct?

9 **A.** Correct.

10 **Q.** And how do you distinguish when you're looking -- when
11 you're just thinking at a high level of dexopt versus the rest
12 of Dalvik, what terminology do you use to distinguish them?

13 **A.** Dexopt runs ahead of time, usually at install time. The
14 operations inside Dalvik are happening at runtime.

15 **Q.** But can we use a term between us to describe that of
16 Dalvik which is not dexopt, the execution or the virtual
17 machine?

18 **A.** Sure, that works.

19 **Q.** Okay. So that was what we were talking about before.

20 And now we're going to turn to dexopt, and I would
21 like to ask you some questions about dexopt. If you look at
22 page -- so this is a document that you wrote to describe the
23 dexopt functions, correct?

24 **A.** Yes.

25 **Q.** If you turn to Page 2 it states:

1 "Dexopt. We want to verify and optimize
2 classes in the DEX file."

3 Correct?

4 **A.** Let me find that.

5 **Q.** Actually, it's on Page 3 under "dexopt." See it says:

6 "Dexopt. We want to verify and optimize all
7 of the classes in the DEX file."

8 Do you see that?

9 **A.** I do.

10 **Q.** And that's correct? That's a correct statement?

11 **A.** Yes.

12 **Q.** The way this works is the dx tool is on the developer
13 platform, and it translates Java bytecode into Dalvik bytecode;
14 correct, sir?

15 **A.** Yes.

16 **Q.** And then dexopt performs optimization on the Dalvik
17 bytecode that was sent to the device after it went through the
18 dx tool?

19 **A.** I agree with most of that.

20 **Q.** After the dx tool translates Java bytecode into the Dalvik
21 byte code, dexopt performs optimizations of the Dalvik
22 bytecode?

23 **A.** Yes.

24 **Q.** All DEX files are run through dexopt?

25 **A.** Yes.

1 Q. Every application that runs on the device has gone through
2 dexopt?

3 A. Correct.

4 Q. And if dexopt does not successfully run an application on
5 a user device, the app will not be installed, correct?

6 A. Yes.

7 Q. The application --

8 **THE COURT:** Double negative problem. You say, "The
9 app will not be installed, correct?" "Yes."

10 So I urge you to be mindful of the double negative
11 problem. So it's up to you if you want to go back over that
12 one.

13 **MR. JACOBS:** Sure.

14 **BY MR. JACOBS:**

15 Q. If dexopt fails, the app will simply not be installed?

16 A. Correct.

17 Q. The application installer causes dexopt -- the application
18 installer causes dexopt to process the DEX files?

19 A. In some situations, yes.

20 Q. On a production device it's the application installer that
21 causes dexopt to process the DEX file?

22 A. For newly installed applications, that is a true
23 statement.

24 Q. For already-installed applications where dexopt has to
25 run, something else causes that to happen?

1 A. Yes.

2 Q. And that case is the case where, for example, there is a
3 system update on the device?

4 A. That is one example, yes.

5 Q. What are the other examples of when dexopt will run aside
6 from initial installation?

7 A. On a production device there are -- there is one, one
8 other primary situation, which is the -- what you can refer to
9 as the firmware on the device. Dexopt will be run for the
10 initial installation, so what you get when you buy the phone,
11 and then also, as you said, for any system updates that happen.

12 On a non-production device it may also be run just
13 kind of as needed. That is typically not even used with an
14 engineering anymore, but it is still possible.

15 Q. By "production device" you mean a phone that an end user
16 would have in his or her hand?

17 A. Yes.

18 Q. If you look at the next sentence that's highlighted in the
19 paragraph that we are look at under "dexopt," it says:

20 "The easiest and safest way to do this is to
21 load all the classes into the VM and run
22 through them."

23 Do you see that?

24 A. I do.

25 Q. And that's a correct statement?

1 A. Yes.

2 Q. And so the dexopt loads the DEX file, the application DEX
3 file into the Dalvik VM, correct?

4 A. Yes.

5 Q. And so it processes -- dexopt processes the DEX files when
6 the Dalvik Virtual Machine is running?

7 A. Yes.

8 Q. And, in fact, you described dexopt in this document as
9 really just a backdoor into the VM, correct?

10 A. Yes.

11 Q. And that's highlighted here on the screen.

12 "The solution is to invoke a program called
13 dexopt, which is really just a backdoor into
14 the VM."

15 Do you see that?

16 A. I do.

17 Q. And you wrote that; correct, sir?

18 A. Yes.

19 Q. And it is an accurate statement?

20 A. Yes.

21 Q. Now, dexopt is important for execution speed?

22 A. I'm struggling with the word "important" just because of
23 the magnitude. It does improve.

24 Q. Well, let's look at the first page of this document. And
25 if I look at the sixth bullet down, this is in a discussion of

1 dexopt and you wrote:

2 "Bytecode optimization. Quickened
3 instructions (method pruning) is important
4 for speed and battery life."

5 Do you see that?

6 **A.** I do.

7 **Q.** And that's a true statement?

8 **A.** It was true when it was written.

9 **Q.** And when was that, sir?

10 **A.** The copyright on the last page is 2008 and that sounds
11 about right.

12 **Q.** Dexopt can run in several modes, one of which is to verify
13 all classes and optimize verified classes; true?

14 **A.** True.

15 **Q.** For the Google G1 device, Google used the mode for
16 verifying, quote, all classes and optimize verified classes,
17 close quote; true?

18 **A.** Yes.

19 **Q.** And used the same mode for the next assess?

20 **A.** I believe so.

21 **Q.** And those were phone -- those were Google co-branded
22 phones?

23 **A.** I think so, yeah.

24 **Q.** Based on dexopt's performance benefits -- true or false.

25 OEMs would likely run dexopt in the mode for verifying, quote,

1 all classes and optimize verified classes, close quote?

2 **MR. KAMBER:** Objection, foundation.

3 **THE COURT:** Overruled. Please answer.

4 **A.** I'm sorry. Could you repeat the first part of that
5 question again?

6 **BY MR. JACOBS:**

7 **Q.** Based on dexopt's performance benefits, would you agree
8 with me that OEMs would likely run dexopt in the mode for
9 verifying all classes and optimize verified classes?

10 **A.** I doubt that performance benefits had anything to do with
11 it. I expect they just left it in the default mode.

12 **Q.** And that is the default mode?

13 **A.** Yes.

14 **Q.** Now, is the application installer itself written in Java?

15 **A.** Yes.

16 **Q.** And it runs on a Dalvik Virtual Machine?

17 **A.** Yes.

18 **Q.** Google makes production builds of Android?

19 **A.** We do.

20 **Q.** And you call that building a system?

21 **A.** Yes.

22 **Q.** And when Google builds that system, it runs dexopt over
23 system files?

24 **A.** Yes.

25 **MR. JACOBS:** May I?

1 **THE COURT:** Yes.

2 (Whereupon, document was tendered
3 to the witness.)

4 **BY MR. JACOBS:**

5 **Q.** 27, more snippets.

6 These are your snippets from January 8th, 2007
7 through January 2, 2008.

8 **A.** Okay.

9 **Q.** Is that true?

10 **A.** Yeah, looks like.

11 **MR. JACOBS:** Offer 27, your Honor.

12 **MR. KAMBER:** No objection.

13 **THE COURT:** Received.

14 (Trial Exhibit 27 received
15 in evidence)

16 **BY MR. JACOBS:**

17 **Q.** Please take a look at the entry on Page 8 for May 14,
18 2007. It says:

19 "Enabled instruction rewriting in the dex
20 optimizer, improves the speed of virtual
21 method calls and access to instance fields."

22 Do you see that?

23 **A.** Yes.

24 **Q.** And so that entry records the date that you enabled
25 dexopt's rewriting of virtual method calls in access to -- and

1 accesses to instance fields?

2 **A.** Yes.

3 **Q.** And there you're referring to improving the speed of
4 virtual method calls?

5 **A.** Yes.

6 **Q.** And you refer to improving the speed of access to instance
7 fields?

8 **A.** Yes.

9 **Q.** Dexopt improves the speed of virtual method calls by
10 replacing the method index with a vtable index?

11 **A.** Yes.

12 **Q.** And dexopt improves the speed of accesses to instance
13 field by replacing the field index with a byte offset?

14 **A.** Yes.

15 **Q.** And you agree that a vtable offset for a method is a
16 specific pointer to a location in memory for data; true, sir?

17 **A.** True.

18 **Q.** And you agree that a byte offset is the same thing; true,
19 sir?

20 **A.** Yes.

21 **Q.** The field index is not itself the location in memory where
22 the instance field is stored?

23 **A.** True.

24 **Q.** Replacing constant pool indices with byte offsets in the
25 bytecode instructions makes the Dalvik VM run faster?

1 A. Possibly.

2 Q. Well, you've measured performance improvements from
3 symbolic reference resolution in the Dalvik VM runtime?

4 A. I have.

5 Q. And you measured about a 20 percent improvement in speed?

6 A. Are you referring to something specific?

7 Q. Let me show you.

8 MR. JACOBS: May I?

9 THE COURT: Yes.

10 (Whereupon, document was tendered
11 to the witness.)

12 BY MR. JACOBS:

13 Q. 258 is an email from you had to colleagues dated May 2,
14 2007; yes?

15 A. Yes.

16 MR. JACOBS: Offer 258 into evidence.

17 MR. KAMBER: No objection, your Honor.

18 THE COURT: It will be received.

19 (Trial Exhibit 258 received
20 in evidence)

21 BY MR. JACOBS:

22 Q. So in this email you're reporting on the results of your
23 optimization work; true?

24 A. Yes.

25 Q. And you reported in summary, short version:

1 "The stuff that you would expect to get
2 faster got faster by about 20 percent. Most
3 tests remained the same, which wasn't the
4 case on all runs (e.g., add test was hanging
5 around 2,000 rather than 1987 [sic] in some
6 earlier trials."

7 Do you see that?

8 **A.** I do.

9 **Q.** And the stuff that you expected to get faster got faster
10 by about 20 percent?

11 **A.** True.

12 **Q.** Let's go back to 739. And if you look down at the bottom
13 of the page -- of the page that's marked at the bottom Page 3
14 of 5?

15 **A.** Okay.

16 (Document displayed)

17 **Q.** Now, there is a paragraph there under "Optimization" that
18 says:

19 "Virtual machine interpreters typically
20 perform certain optimizations the first time
21 a piece of code is used. Constant pool
22 references are replaced with pointers to
23 internal data structures."

24 Do you see that?

25 **A.** I do.

1 Q. It goes on to say:

2 "Operations that always succeed or always
3 work a certain way are replaced with simpler
4 forms. Some of these require information
5 only available at runtime, others can be
6 inferred statically when certain assumptions
7 are made."

8 Do you see that?

9 A. I do.

10 Q. You wrote that paragraph?

11 A. Yes.

12 Q. Some of the information -- and that's about dexopt?

13 A. Well, it's about optimization in general.

14 Q. Well, this paragraph is under the -- under "dexopt," isn't
15 it, sir?

16 A. This is a dexopt document.

17 Q. And so some of the information that dexopt requires is
18 only available at runtime; true, sir?

19 A. No.

20 Q. So it's your testimony that dexopt does not require any
21 information only available at runtime?

22 A. Yes.

23 Q. Now, the dx tool is part of the Android SDK?

24 A. I believe so.

25 Q. It's used by developers to convert compiled Java class

1 files to Android DEX files?

2 **A.** Yes.

3 **Q.** Would you agree that developers run the dx tool at compile
4 time rather than runtime?

5 **A.** Yes.

6 **Q.** So that means you could have had the dx tool perform all
7 the optimizations performed by dexopt since, by your testimony,
8 those optimizations can run at compile time?

9 **A.** You could, but we chose not to.

10 **Q.** And the reason you chose not to is because there is
11 information that you need in order to run dexopt that is
12 required -- that you need to get from a device on which the
13 application is being installed?

14 **A.** Yes.

15 **Q.** And the dx tool -- back to the dx tool for a minute.

16 It outputs instructions containing instant field
17 indices rather than byte offsets for fields; true?

18 **A.** True.

19 **Q.** And the dx tool cannot resolve the symbolic references
20 because it doesn't know where in a particular Android device's
21 memory the data will actually be stored?

22 **A.** True.

23 **Q.** Dexopt is something that can only be done if the device's
24 architecture and device's operating system is known so that the
25 byte offset can be calculated?

1 A. I'm struggling a little bit with making it about the
2 computer architecture.

3 Q. How about the memory layout?

4 A. I like that better.

5 Q. So dexopt is something that can only be done if the memory
6 layout in the device's operating system is known so that the
7 byte offset can also be known?

8 A. I don't believe a device's operating system must be known.

9 Q. But the device's memory layout must be known?

10 A. Yes.

11 Q. And that information must be gathered by dexopt as it's
12 doing it's optimization on the device itself?

13 A. No.

14 Q. How is that information obtained?

15 A. The information comes from the set of -- we call them
16 bootstrap classes. So they are a set of classes that are
17 available on the device. And then the virtual machine itself
18 has a specific idea of how it wants to lay out fields and
19 methods within a class.

20 Q. Now, the reason you need to run dexopt when you do a
21 system update is because the memory layout would change?

22 A. It could change.

23 Q. And you need to make sure that it hasn't changed, so you
24 run dexopt when there is a software system upgrade; true, sir?

25 A. Yes.

1 Q. And by the software system now, we're not referring to the
2 memory hardware; we're referring to software that's installed
3 on the device?

4 A. Yes.

5 Q. And because that software installed on the device can
6 affect the memory layout that dexopt depends on?

7 A. Yes.

8 Q. True or false: The resolving functions run by Dalvik VM
9 and dexopt work similarly in that if the resolution has not
10 been done before, they perform the resolution and store the
11 resulting pointer in the resolved items table?

12 A. True.

13 Q. And, in fact, dexopt calls `dvmOptResolveClass` in
14 `Optimize.c` to resolve classes?

15 A. Yes.

16 Q. And `dvmOptResolveClass` takes `classIdx` as an input?

17 A. I believe so.

18 Q. And assuming no errors for non-primitive type,
19 `dvmOptResolveClass` calls `dvmDexGetResolvedClass`,
20 `dvmFindClassNoInit`, and `dvmDexSetResolvedClass`; correct?

21 A. Yes.

22 Q. So now let's look at `Resolve.c`, which is back on your
23 table there, I believe, at 47.6.

24 A. I have it.

25 Q. And if you turn to Page 2, Line 63 the Dalvik VM calls

1 dvmResolveClass in Resolve.c to resolve classes, correct?

2 **A.** Yes.

3 **Q.** And dvmResolveClass takes classIdx as an input?

4 **A.** Yes.

5 **Q.** And assuming no errors for a non-primitive time

6 dvmResolveClass calls dvmDexGetResolvedClass,

7 dvmFindClassNoInit and dvmGetSetResolvedClass?

8 **A.** Yes.

9 **Q.** So Resolve.c and Dexopt.c are calling the same functions
10 to perform resolution?

11 **A.** I assume you mean Optimize.c. You said Dexopt.c.

12 **Q.** Yes, optimize.c.

13 **A.** In that case, yes.

14 **Q.** To clear up one small piece of this, when we said, "See
15 DVM," when we see the initials "DVM," that's a reference to
16 Dalvik Virtual Machine?

17 **A.** Yes.

18 **MR. JACOBS:** Can we put up Mr. Van Nest's opening
19 slide 35 from his opening argument?

20 (Document displayed)

21 **BY MR. JACOBS:**

22 **Q.** This is a slide that we all saw, it seems like ages ago,
23 but I think it might have been just yesterday.

24 And you've seen this slide before, right, sir?

25 **A.** I have.

1 Q. And in this slide 01 in the instruction is an index to a
2 location in the field ID table, true?

3 A. Yes.

4 Q. And then on the far right there's the word "fun." Do you
5 see that?

6 A. I do.

7 Q. And the word "fun" is a variable; true, sir?

8 A. It's a constant.

9 Q. It's not the data you're looking for; true, sir?

10 A. I guess that depends on who "you" is in that sentence.

11 Q. Fun is a constant, but fun doesn't -- in this location in
12 string data, there is no value for fun; true, sir?

13 A. I don't understand what you're asking.

14 Q. Tell us again what the word "fun" is?

15 A. It is a string constant, which in this case I believe is
16 intended to be the name of a field.

17 Q. And insofar as the program might be looking for the
18 value -- the data value associated with that field, would it
19 find that data value in the position 03 under string data?

20 A. No.

21 Q. No further questions.

22 **THE COURT:** All right. Let's have cross-examination.

23 **CROSS EXAMINATION**

24 **BY MR. KAMBER:**

25 Q. Good morning, Mr. McFadden.

1 A. Good morning.

2 Q. I just want to ask you a few questions to clear up what
3 may be some confusion with the terminology.

4 Mr. Jacobs began his questioning by talking about dex
5 code and whether you would agreed that that is the same thing
6 as Dalvik bytecode instructions; do you remember that?

7 A. I do.

8 Q. Okay. Where were you -- why do you prefer to use the term
9 "Dalvik bytecode instructions"?

10 A. Well, DEX is the file format. So it's a little strange to
11 refer to it as DEX code.

12 Q. When you were talking about dex code, were you assuming
13 that Mr. Jacobs was talking about the Dalvik bytecode
14 instructions within a DEX file?

15 A. Yes.

16 Q. Are Dalvik bytecode instructions the only thing in a DEX
17 file?

18 A. No.

19 Q. What other types of things are in a DEX file?

20 A. Lots of strings, lots of indexes that point to other
21 things, offsets; lots and lots of data.

22 Q. Okay. Are the Dalvik bytecode instructions contained in a
23 separate distinct part of a DEX file?

24 A. Yes.

25 Q. And are there symbolic references in a different part of

1 that DEX file?

2 **A.** Yes.

3 **MR. JACOBS:** Objection, your Honor.

4 **MR. KAMBER:** Your Honor, he asked -- Mr. Jacobs asked
5 if there were symbolic references in the DEX file. I'm just
6 asking the same questions using the same terminology.

7 **THE COURT:** Overruled. Please answer.

8 **A.** Yes.

9 **BY MR. KAMBER:**

10 **Q.** Okay. Are the symbolic references in the Dalvik bytecode
11 instructions?

12 **A.** No.

13 **MR. KAMBER:** Now, Ben, can we pull up TX 739 very
14 briefly?

15 (Document displayed)

16 **MR. KAMBER:** Actually, take that down for a moment.

17 **BY MR. KAMBER:**

18 **Q.** I just want to ask one more question about the
19 instructions.

20 Are the indexes in the instructions, in the Dalvik
21 bytecode instructions, in the dex file?

22 **A.** Yes.

23 **Q.** Are indexes numeric references?

24 **A.** Yes.

25 **Q.** Why are indexes numeric references instead of symbolic

1 references?

2 **A.** Symbolic references require matching two pieces. Numeric
3 references just -- it's just an index into a table. It
4 identifies the location of the data in and of itself.

5 **Q.** Now let's go to TX 739.

6 There was some questioning about the comments in the
7 section labeled "Optimization." It starts at the bottom of
8 page 3, Mr. McFadden, and continues on to 4.

9 **A.** Right.

10 **Q.** I believe there may be some confusion with what this
11 document says. Mr. Jacobs asked you some questions regarding
12 this sentence:

13 "Some of these require information only
14 available at runtime. Others can be inferred
15 statically when certain assumptions are
16 made."

17 Do you see that?

18 **A.** Yes.

19 **Q.** What did you mean when you wrote that?

20 **A.** Uhm, there are things that you can only do at runtime.
21 For example, if you do an optimization that requires knowing
22 which methods are used a lot. That isn't something that you
23 can figure out just by looking at the file.

24 Statically -- static optimizations you can do just --
25 you look at the file, you see some things that maybe you can

1 take a shortcut here and there. And that's what dexopt does.

2 **Q.** Dexopt does static optimizations; is that correct?

3 **A.** Yes.

4 **Q.** So when it's referring here in the sentence, "Others can
5 be inferred statically when certain assumptions are made," what
6 is that referring to?

7 **A.** Uhm, well, the bullet points that follow describe the
8 various optimizations that dexopt can do.

9 **Q.** Are you distinguishing here between the optimizations that
10 dexopt does statically at install time, for example, with
11 optimizations that can be performed only with runtime
12 information?

13 **A.** Yes.

14 **Q.** Okay. Does dexopt require any runtime information from
15 the virtual machine in order to operate?

16 **A.** No.

17 **MR. KAMBER:** Your Honor, I would reserve the rest of
18 our questioning for our case-in-chief. We'll be putting on
19 Mr. McFadden either later today or tomorrow.

20 **THE COURT:** All right. I'll let you do that.

21 Any further questions, Mr. Jacobs?

22 **MR. JACOBS:** Could we have 35 up again, please. I'm
23 sorry, the opening slide.

24

25

REDIRECT EXAMINATION

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

BY MR. JACOBS:

Q. Now, in the instructions in dex code -- we saw some examples of this earlier -- there's this "field@CCCC." That's an index, right?

A. Yes.

Q. And there's a reason that instructions of -- in programming that you use a field index as opposed to a variable like "fun"; isn't there, sir?

A. Yes.

Q. And the reason is that the instructions have a set format; correct, sir?

A. They have a -- I'm not sure what you mean by "a set format."

Q. Well, "fun" -- you see "fun" could be fun or it could be funny or it could be funnily. You only have a certain number of positions in the instruction to take advantage of; correct, sir?

A. Uhm, okay. I see where you're going.

Yes. Dalvik uses fixed-width instructions.

Q. Thank you. Fixed-width instructions.

So the -- because of fixed-width instructions, and because of a constant like "fun" could be fun, funny, funnily, it could be a variable length, you have to use an index in the in-line instructions; correct, sir?

1 A. Yes.

2 Q. And so in this -- in this chart, Exhibit 35, what we're
3 seeing is that having had to use an index, we have to chase
4 this chain in order to get to the constant "fun"; true, sir?

5 A. Yes.

6 Q. Having chased the chain to get to the constant "fun,"
7 "fun" is a symbolic reference; true, sir?

8 A. True.

9 Q. So we've chased the chain from a field index to a symbolic
10 reference, not to a numeric reference; true, sir?

11 A. True.

12 Q. So it's your position, your testimony on examination just
13 a minute ago, that because the field index is a pointer to a
14 symbolic reference, it is a numeric reference?

15 A. Yes.

16 Q. That's your view?

17 A. Yes.

18 MR. JACOBS: All right. Thank you very much.

19 THE COURT: All right. May the witness step down?

20 MR. JACOBS: Yes, subject to --

21 MR. KAMBER: Yes. We will be recalling him.

22 THE COURT: Mr. McFadden, thank you. You are free to
23 go. Leave all of our documents here, please.

24 We will now call our next witness.

25 (Witness steps down.)

1 through the announcement of the acquisition of Sun by Oracle, I
2 was the executive vice president of corporate development and
3 alliances.

4 Q. Then after the acquisition by Oracle of Sun, did you
5 continue to work at now Oracle?

6 A. Yes, I did.

7 Q. And what was your position at Oracle?

8 A. I was senior vice president, CEO office.

9 Q. When did you -- and you left Oracle?

10 A. I did.

11 Q. When was that?

12 A. In January of this year.

13 Q. And are you working now, or are you taking a break?

14 A. I am unemployed. Taking a break.

15 Q. Were you a member of something called the Executive
16 Leadership Team at Sun?

17 A. Yes, I was.

18 Q. And when was that -- what was the period in which you were
19 on that team?

20 A. I think we had the group called the ELT, started in -- I
21 think it was 2004.

22 Q. And then it lasted through the announcement of the
23 acquisition?

24 A. Yes.

25 Q. And was -- can you describe the kinds of meetings and

1 discussions that took place in the ELT?

2 **A.** Yes. The ELT was, essentially, the CEO staff. So it was
3 the senior executives of the company. Consisted of 10 or 12
4 people. And we had regular meetings to discuss important
5 issues for the company: business strategy, operational issues.

6 **Q.** Can you describe the kind of relationship you had with the
7 CEO, Jonathan Schwartz, during your period -- the period when
8 you were a member of the ELT?

9 **A.** Yes. I was -- in fact, I was actually very close to
10 Jonathan.

11 Given my role for both the mergers and acquisitions
12 and also the strategic alliances, I had responsibility for a
13 lot of the issues that were really, really important to
14 Jonathan and other members of the ELT. So I spent a lot of
15 time with Jonathan.

16 **Q.** And the time you spent with him, was that only in ELT
17 meetings?

18 **A.** Oh, no. His office was just a few doors away from mine.
19 So he would drop in regularly, sometimes several times a day,
20 just to talk about things on his mind.

21 **Q.** What discussions do you recall at the ELT regarding
22 Android? Do you recall discussions at the ELT regarding
23 Android?

24 **A.** Yes, I do.

25 **MR. VAN NEST:** Objection. Hearsay, Your Honor.

1 **MR. JACOBS:** State of mind, Your Honor.

2 **MR. VAN NEST:** State of mind is irrelevant. It's
3 hearsay.

4 **THE COURT:** Overruled. But this is not going to be
5 admitted for the truth. It will be admitted only to prove up
6 transactions, meaning conversations as to what was said or not
7 said on particular subjects. So it's admitted for that limited
8 purpose.

9 Go ahead. Overruled.

10 **BY MR. JACOBS:**

11 **Q.** How often did discussions of Android occur?

12 **A.** Uhm, I can't remember exactly, but there were -- it seemed
13 as though the frequency of those discussions increased as ELT
14 members became more aware of the potential that Android was
15 using Java.

16 **Q.** And did you also participate in informal discussions about
17 Android with other members of the Sun management team?

18 **A.** Uhm --

19 **MR. VAN NEST:** Objection, Your Honor. This is
20 hearsay. Informal discussions. They are no longer
21 transactions.

22 **MR. JACOBS:** Just asking about the fact of the
23 discussions right now, Your Honor.

24 **THE COURT:** Overruled. Please answer.

25 **THE WITNESS:** Yes.

1 **BY MR. JACOBS:**

2 **Q.** And did those discussions include informal discussions
3 with Jonathan Schwartz?

4 **A.** Yes.

5 **Q.** At any time did anyone on the leadership team ever express
6 concern -- for the questions I'm going to ask now, I want to
7 separate legal advice from lawyers, and just ask you about ELT
8 members' discussions.

9 At any time did anyone on the leadership team ever
10 express the concern that Sun did not have grounds to pursue
11 Google over its use of Java-related intellectual property in
12 Android?

13 **A.** No, there were no such discussions.

14 **Q.** At any time did anyone on the leadership team ever express
15 concern -- with that same guidance from before -- that Sun's
16 intellectual property claims, if brought against Android, would
17 be weak?

18 **A.** No. There were no such discussions.

19 **Q.** At any time, did Sun make an affirmative decision not to
20 assert IP claims against Google?

21 **A.** No, not that I'm aware of.

22 **Q.** In the spring of 2009, after Oracle announced that it
23 would acquire Sun, did your responsibilities change?

24 **A.** Yes, they did.

25 **Q.** And if so -- and how did they change?

1 **A.** In October of 2009, I was asked to assume additional
2 responsibilities, in addition to -- at the time, I was the --
3 excuse me, the integration lead executive for the acquisition.
4 And the board asked me to expand my responsibilities to assume
5 operating responsibilities for the company overall.

6 **Q.** And when was that latter request made?

7 **A.** That was in October of 2009.

8 **Q.** Now, as a matter of title, did Mr. Schwartz remain CEO?

9 **A.** Yes. His title didn't change.

10 **Q.** And how about his authority?

11 **A.** I think his authority, at that point, was significantly
12 reduced.

13 **Q.** And your authority?

14 **A.** Was significantly increased.

15 **Q.** Did Mr. Schwartz maintain a blog while he was at Sun?

16 **A.** Yes, he did.

17 **Q.** Did you read it?

18 **A.** Usually not.

19 **Q.** Was Mr. Schwartz's blog an official statement of Sun
20 policy or decisions?

21 **A.** No, it wasn't. We had a very clear policy at Sun that any
22 blogs were just personal to the person who posted them, and
23 that they didn't represent any official statements or policies
24 of the company.

25 **Q.** During the transition after the announcement of the

1 acquisition and before the closing, are you aware of the fact
2 of discussions by the Sun legal department with the Oracle
3 legal department about claims against Google relating to
4 Android?

5 **A.** Yes, I am.

6 **Q.** And did this -- these discussions include the topic of
7 patent claims?

8 **A.** Yes.

9 **MR. VAN NEST:** Objection. Hearsay.

10 **THE COURT:** Overruled. The subject matter is not
11 privileged.

12 **MR. VAN NEST:** Lack of foundation, Your Honor.

13 **THE COURT:** Well, overruled.

14 **MR. JACOBS:** No further questions.

15 **THE COURT:** All right. Please, cross-examination.

16 **CROSS EXAMINATION**

17 **BY MR. VAN NEST:**

18 **Q.** Good morning, Mr. Sutphin.

19 **A.** Good morning.

20 **Q.** I take it you and Mr. Schwartz were colleagues at Sun for
21 many years?

22 **A.** Yes, we were.

23 **Q.** You worked together?

24 **A.** Very closely.

25 **Q.** And then in 2006, he became CEO?

1 A. Yes.

2 Q. And was that a position you competed for?

3 A. Not at all.

4 Q. And you began working for him?

5 A. I had been working for him before that.

6 Q. Okay. And so during this period from 2006 to 2010, you
7 were reporting to Mr. Schwartz, not the other way around,
8 right?

9 A. Correct.

10 Q. He was the CEO?

11 A. Yes.

12 Q. He had ultimate responsibility for business decisions at
13 the company?

14 A. Uhm, to -- to a point.

15 Q. And --

16 A. Subject to approval by the board.

17 Q. Right. Subject to approval by the board.

18 Other than the board, he was the highest-ranking
19 official at the company, correct? And you reported to him?

20 A. Yes.

21 Q. He was the ultimate decision-maker on business issues,
22 right?

23 A. Subject to the qualification I mentioned about the board's
24 responsibility for important decisions.

25 Q. And he was the ultimate decision-maker on alliances?

1 A. Yes.

2 Q. Ultimate decision-maker on negotiations with other
3 companies?

4 A. Yes.

5 Q. And you mentioned that he wrote a blog.

6 A. Yes.

7 Q. Were you aware that the company was representing to the
8 government that the CEO blog was an official statement of the
9 company?

10 MR. JACOBS: Objection, Your Honor. Mischaracterizes
11 the representation, the question.

12 THE COURT: You may do this. You may pull out the
13 document, explain -- just read from the document. What
14 document number is this?

15 MR. VAN NEST: This is Exhibit 971. It's in
16 evidence, Your Honor.

17 THE COURT: Right. And you may read the title of the
18 document and to whom it was submitted.

19 MR. VAN NEST: The title of the document is
20 "United States Securities and Exchange Commission, Form 10-K."

21 THE COURT: For who?

22 MR. VAN NEST: For Sun Microsystems, fiscal year
23 ending in June 30, 2008.

24 THE COURT: Turn to the page in question and read
25 exactly the language --

1 **MR. VAN NEST:** Can I -- it's in evidence. Can I
2 display it for the jury, Your Honor?

3 **THE COURT:** Yes, you may. Yes.

4 **MR. VAN NEST:** Let's display 971.

5 (Document displayed.)

6 **MR. VAN NEST:** Can we blow up the middle of the page.

7 **BY MR. VAN NEST:**

8 **Q.** By the way, you're aware, Mr. Sutphin, the 10-K is
9 something that is filed each year by Sun, with the government?

10 **A.** Yes.

11 **Q.** You file it with the Securities and Exchange Commission?

12 **A.** Yes.

13 **Q.** That's a requirement for all American corporations to do
14 that?

15 **A.** Yes.

16 **Q.** It's signed by the chairman of the board, Mr. McNealy?

17 **A.** Yes.

18 **Q.** And signed by the CEO, Mr. Schwartz?

19 **A.** Yes.

20 **Q.** And at page 3 -- you can see it on the screen there -- the
21 first line says:

22 "Our Internet address is <http://www.sun.com>.

23 The following filings are posted to our

24 Investor Relations website located at

25 <http://www.sun.com/investors> as soon as

1 reasonably practicable after submission to
2 the United States Securities and Exchange
3 Commission (SEC)."

4 Do you see that language?

5 **A.** Yes.

6 **Q.** And then a little bit further in the paragraph --

7 **MR. VAN NEST:** Can we make that a little bigger, or
8 no?

9 **UNIDENTIFIED SPEAKER:** No.

10 **MR. VAN NEST:** No. Okay.

11 "We periodically webcast company
12 announcements, product launch events and
13 executive presentations which can be viewed
14 Via our Investor Relations web site."

15 **BY MR. VAN NEST:**

16 **Q.** Did I read that correctly?

17 **A.** Yes.

18 **Q.** (As read:)

19 "Additionally, we provide notifications of
20 our material news including SEC filings,
21 investor events, press releases, and CEO
22 blogs as part of the Official Investor
23 Communications section of our Investor
24 Relations web site."

25 Do you see that --

1 A. I do.

2 Q. -- Mr. Sutphin?

3 A. Yes.

4 Q. Now, was this statement filed each year by the company
5 with the SEC, this -- a 10-K?

6 A. 10-K, yes.

7 Q. And were you aware that the company was representing to
8 the government that the "CEO blog" -- I assume that refers to
9 Mr. Schwartz's blog?

10 A. Yes.

11 Q. -- was an official statement of the company?

12 A. Uhm, I don't read that statement from this language here.

13 Q. Now, I take it -- is it your testimony that you never read
14 Mr. Schwartz's blog?

15 A. No, not "never," but rarely.

16 Q. Were you aware that when Android was released in 2007,
17 Mr. Schwartz said, "Welcome to the Java community. Android has
18 put a rocket onto Java"?

19 A. I'm aware of that blog.

20 Q. So that's one of the blogs that you did read?

21 A. No, I didn't read it, but somebody told me about it.

22 Q. Okay. So you were aware of it at the time?

23 A. Not at the time. It was after the fact.

24 Q. And how soon after the fact did you become aware of the
25 blog?

1 A. I really can't recall. It was quite a while.

2 Q. And did you take any steps to correct the blog?

3 A. No.

4 Q. Did you see anything wrong with the blog?

5 A. I didn't read the blog, so I didn't really have a good
6 sense of exactly what it said.

7 Q. Well, at the time you became aware of it, was it
8 consistent with what you understood to be the policy of Sun at
9 the time?

10 A. Uhm, no, I guess I -- I would have -- I don't recall
11 having -- when I became aware of that blog, I don't recall
12 having a discussion or thinking to myself was this consistent
13 or inconsistent with the policy of the company.

14 Q. Fair enough.

15 Admittedly, you didn't go running into Mr. Schwartz'
16 office, which was right next to yours, and saying, Jonathan,
17 what have you done? Right?

18 A. No.

19 Q. And you didn't bring up, in executive committee or
20 anywhere else, this blog, right?

21 A. Correct.

22 Q. Now, have you ever heard Mr. Schwartz say publicly, as
23 CEO, that the policy of Sun is to innovate, not litigate?

24 A. Yes. Something to that effect.

25 Q. And that's something that he stated publicly on more than

1 one occasion?

2 **A.** Most likely.

3 **Q.** Have you heard him say it in group meetings?

4 **A.** Uhm, I don't know if he said those exact words, but that
5 was certainly the attitude of the company, that we preferred
6 not to litigate.

7 **Q.** And to innovate, not litigate, correct?

8 **A.** That's correct.

9 **Q.** Did you also hear Mr. Schwartz say on numerous occasions
10 that, "We only use our patents for defensive purposes, not
11 offensive purposes"?

12 **A.** I don't know if I have heard him say that.

13 **Q.** Do you remember reading any blog posting or other company
14 statement to the effect that, "We use patents to protect
15 ourselves for defense, but not for offense"?

16 **A.** Uhm, yes.

17 **Q.** And was that said on more than one occasion by
18 Mr. Schwartz or other members of Sun's executive team?

19 **A.** Uhm -- you know, I'm not sure how to answer the question
20 because I don't remember specific statements made by people to
21 that effect.

22 What I do remember is that the company kind of
23 culture was that we preferred not to be the plaintiff in
24 litigation over intellectual property. But having said that,
25 there were many exceptions to that where we felt like we needed

1 to protect our rights.

2 **Q.** But, in any event, Mr. Sutphin, on at least one occasion
3 and maybe more public statements by Mr. Schwartz and others
4 were to the effect that, we use our patents for defense, not
5 offense, that's what we prefer, right?

6 **A.** Yes, I think so.

7 **MR. VAN NEST:** I have Nothing further, Your Honor.

8 **REDIRECT EXAMINATION**

9 **BY MR. JACOBS:**

10 **Q.** Mr. Sutphin, on 971 Mr. Van Nest asked you about a
11 reference to blog posts. Do you recall that?

12 **A.** Yes.

13 **Q.** And were all of Mr. Schwartz's blog posts on matters of
14 substance to the company's business?

15 **A.** No.

16 **Q.** Can you give an example of one that clearly was not part
17 of the -- a matter of the company's business?

18 **A.** Uhm, sure. There was one where he talked about an April
19 Fools prank that was played on him.

20 **Q.** Does the sentence that Mr. Van Nest asked you about,
21 "additionally, we provide notice (sic) of our material news,"
22 and then says "and CEO blogs," did you understand that to be
23 referring to every CEO blog, or some CEO blogs?

24 **A.** Some CEO blogs.

25 **MR. JACOBS:** Thank you, sir.

1 **THE COURT:** All right.

2 **MR. VAN NEST:** Your Honor, we did agree that
3 Mr. Bornstein could be called in plaintiff's case.

4 As Your Honor is aware, he is not available until
5 tomorrow. So I think it would be fair to say he's also a
6 witness in the plaintiff's case.

7 **THE COURT:** But other than that, is Professor
8 Mitchell your last witness on your case-in-chief?

9 **MR. JACOBS:** Yes.

10 **THE COURT:** All right. I just wanted to -- sometimes
11 it helps to let the jury know where we are. So, apparently, we
12 have two more witnesses, the witness on the stand, and then
13 Mr. Bornstein. But he can't come, so we are making progress.
14 And we're on track.

15 And so now we have Professor Mitchell back, and you
16 will all remember him. Thank you.

17 Go right ahead.

18 **MR. VAN NEST:** Excuse me, Your Honor. You said
19 Mr. Bornstein can't come. He can't come today. He will be
20 here tomorrow.

21 **THE COURT:** That's what I meant. He will be here
22 tomorrow, I guess. Okay.

23 **MR. VAN NEST:** Excuse me.

24 **MR. JACOBS:** Your Honor, may I hand Dr. Mitchell the
25 slides he's going to be using in his report?

1 **A.** I studied the patents to understand them, other material
2 that was made available to me as part of the legal proceedings
3 here. I also looked at Android source code.

4 I installed the development kit to build my own
5 versions of Android. I modified the software, installed
6 builds. Ran it.

7 Also examined phones using the software in the SDK to
8 examine software on the phone and look at that, also.

9 **Q.** Did you also look at some Google documents apart from the
10 code itself?

11 **A.** Yes. There's Google documentation on the Web and in other
12 forms that was provided to me.

13 **Q.** What versions of Android did you review?

14 **A.** I looked primarily at the Froyo and Gingerbread versions,
15 but I also did examination of differences in the code across
16 the versions, starting from 1.1 up through those versions.

17 **Q.** And Froyo, that's 2.2; and Gingerbread is 2.3?

18 **A.** I believe so, yes.

19 **Q.** And the infringement evidence that you are going to be
20 going through with the jury today is taken from which versions
21 of Android?

22 **A.** Primarily, Froyo and Gingerbread. I think the slides show
23 those two versions.

24 **Q.** And you've looked at the other versions?

25 **A.** Yes. I've run programs that compute differences between

1 different software bases to examine the changes across
2 different versions of the software.

3 Q. And as to the other versions that preceded 2.2 and 2.3,
4 were there differences relevant to your infringement analysis?

5 A. No.

6 Q. You referred to some experiments you conducted. What --
7 what experiments did you conduct on the Android SDK or Android
8 devices?

9 A. Using the SDK and an emulator and the phones, one
10 experiment I did was take the source code and add print
11 statements, in effect to see which functions are being called
12 as the system runs; rebuild that and run it under the emulator,
13 to see how the system worked and to make sure that functions
14 relevant to my analysis are called.

15 I also looked at using the same software and a laptop
16 computer. Connected the laptop to particular phones and was
17 able to transfer software off the phone onto the laptop, and
18 look at the software on the laptop to see what functions are
19 present in the software on the phone.

20 I also was, at some point later, given access to some
21 Gmail source code. That's the Google e-mail application that
22 runs on the phone. Was able to compare the Gmail source code
23 and some instructions there with instructions on the phone to
24 see that some transformations associated with the '520 Patent
25 took place.

1 Q. Now, you're aware, from being in the courtroom or being
2 apprised of Google's positions in this case, that Google says
3 that it distributes the source code that you have focused on
4 for your infringement analysis, but says it doesn't know if
5 third parties modify the code?

6 A. Yes.

7 Q. Do you have any reason to believe that third parties
8 modify the Android source code that you are going to walk us
9 through?

10 A. No, I do not. In fact, I've seen "read me" files and
11 other things on Samsung and other sites suggesting the
12 contrary.

13 Q. And the extent you didn't actually review source code for
14 a specific phone, do you have a reason for believing that the
15 manufacturers would not modify the portions that you have
16 focused your analysis on?

17 A. Yes.

18 Q. And what are those reasons?

19 A. One reason is there's a Compatibility Test Suite. This is
20 a relatively extensive set of programs that are run to check
21 that an Android software build is consistent with and
22 compatible with the standard Android setup.

23 This Compatibility Test Suite is produced by and
24 distributed by Google. And it's used by vendors and others to
25 test compatibility with that. I did run this. It runs for

1 about eight hours. It does a lot of checks. I was able to see
2 from the code that I instrumented how extensive it was. And I
3 understand and I believe that's used by vendors to check
4 compatibility.

5 **Q.** Are there other reasons why phone makers would be unlikely
6 to modify the relevant portions of Android when they install it
7 on their devices?

8 **A.** Well, the functionality we will talk about further today,
9 that's associated with these patents, provides a very
10 significant performance improvement.

11 It seems unlikely that a phone manufacturer would
12 remove something that's very effective and helpful for their
13 customers.

14 **Q.** Now, we have heard about some modifications that phone
15 makers make to the software that runs on their phones. Are
16 those modifications that we've heard about directed to the
17 portions of the code that you have focused on for your
18 analysis?

19 **A.** I don't believe so. Mostly, I think phone manufacturers
20 want their name to show up when you power on the phone, and
21 some other things of that sort.

22 **Q.** So let's dive in.

23 **MR. JACOBS:** Can we have the slides up, Mr. Lee.

24 (Document displayed.)

25

1 **BY MR. JACOBS:**

2 **Q.** So let's just set the stage for the discussion that
3 follows, Professor Mitchell.

4 Can you explain what you're illustrating on your
5 slide 1.

6 **A.** This is a slide that maybe we've all seen before. This
7 shows Java developers on the left and Android application
8 developers on the right.

9 And the way that Java supports this write once, run
10 anywhere feature allows the Java-developed applications to run
11 on any compatible Java Virtual Machine.

12 Although there are some similarities in the way that
13 Android applications are built, the actual Android applications
14 are designed and run on an Android Dalvik Virtual Machine,
15 which is not the same thing.

16 **THE COURT:** May I remind the jury, I think by now you
17 know that these illustrative slides do not come into evidence.
18 You will not have them in the jury room.

19 So if there's something there that appeals to you,
20 you ought to make a note. Or if there's a question you have,
21 you ought to make a note. Because these slides won't be in the
22 jury room.

23 Nor will any of the expert reports be in the jury
24 room. They are hearsay unless both sides were to somehow
25 agree, which almost never happens in litigation, on expert

1 reports. So you will hear it just from the -- from the words
2 that are verbalized here in the courtroom. That's the
3 evidence.

4 So it's the words coming from Professor Mitchell that
5 will be the evidence. It's not going to be anything in
6 writing, from this witness anyway. And that's going to be true
7 for the corresponding expert witness on the other side.

8 So I say this to you, in case you want to take notes.
9 All right. You don't have to. It's up to you.

10 Thank you. Go ahead.

11 **MR. JACOBS:** Thank you, Your Honor.

12 **BY MR. JACOBS:**

13 **Q.** Let's look at your slide 2, Professor Mitchell. What is
14 this illustrating?

15 **A.** This is the Java platform components that are relevant to
16 the discussion we'll have.

17 At the top, Java application source code written by
18 programmers is written in the Java Language. That is compiled
19 using the Java compiler. So those two steps are for the
20 developer and the developer environment.

21 After that, the Java compiler outputs bytecode, and
22 then that can be used on a device such as a phone or a server
23 or a laptop computer or a desktop computer.

24 The platform where this is executed includes a Java
25 virtual machine that runs Java bytecode, and computing device

1 hardware and other components that are used in computing.

2 **Q.** So now let's compare that to the basic platform components
3 for Android. What are the similarities and differences?

4 **A.** At this outset, you can see that there's both a developer
5 phase, some engineers sitting in an office or cafe, wherever
6 people develop software these days. And the components in the
7 development environment are source code compiled.

8 And then the Java bytecode produced by the Java
9 compiler -- same Java compiler as in the Oracle picture to the
10 left -- is then run through something called the dx tool that
11 converts Java bytecode to dex code.

12 Then on the phone or other device there's a Dalvik
13 Virtual Machine that executes the dex and Dalvik code over that
14 platform.

15 So they are parallel in many ways. And, in
16 particular, there's a distinction between the build time,
17 development time, compile time at the top of the slide, you
18 know, and the runtime execution and things that occur on the
19 platform phone.

20 **Q.** Now, there are some differences, as you described. Are
21 those differences sufficient in and of themselves to lead to
22 some kind of inference that Dalvik does not infringe -- or
23 Android does not infringe the '104 and '520 Patents?

24 **A.** No. There are a number of similarities, and I think we'll
25 get into it, which will show the correspondence. Show the

1 correspondence.

2 **Q.** Now, what are you illustrating on slide 4?

3 **A.** Just to be clear about some of these terms, where names
4 occur in source code and bytecode, I prepared a slide where
5 there's a sample source code.

6 The Java source code at the top has a field name
7 called "test1." And in the source code, the field name "test1"
8 is assigned the data, the string, "HelloWorld!"

9 That source code is compiled by a developer engineer,
10 using the Java compiler, into Java bytecode. And you can see
11 in the Java bytecode -- this is just a small amount of it --
12 where the test1 field name is represented in the bytecode.

13 The test1 field name is a constant. And it's stored
14 in the constant pool. And it's constant #3 in this particular
15 case.

16 As the bytecode is transformed into dex code, we can
17 see some of the same structure in this very small excerpt of
18 the larger dex file. This particular field name is the 0
19 field.

20 Computer scientists often count 0123 when they number
21 the elements of a list, instead of 1234. So this is at the
22 beginning of the array or list of field IDs. And the 0 element
23 in this field ID is the field name "test1." It has a class and
24 a type and a name, which is a string. This string is actually
25 the 14th string. And that's what the number 1400 name_idx

1 means.

2 Q. So when I was asking Mr. McFadden about some of his work,
3 we talked about class_idx, type_idx, name_idx. You were here
4 for that?

5 A. Yes.

6 Q. And you tie back what you're showing here to what
7 Mr. McFadden was saying?

8 A. So he described this fairly extensively. Test1 is a
9 symbolic reference. And the symbolic reference in the dex code
10 is found by looking at this field Ids array. And then
11 following the index 14, if you actually need the letters
12 t-e-s-t-1.

13 Q. What is class_idx, type_idx, and name_idx?

14 A. Those are also indexes into other parts of the file.
15 There's a class_idx array that's a list of classes. And this
16 would -- this is the -- this here indicates that the test1
17 field is a field of objects of class number 0. And the
18 type_idx is used similarly.

19 Q. What happens in the process of creating an application for
20 an Android device? We're looking at slide 5 now.

21 A. Yes. So this slide is just -- adds a little additional
22 information to the previous slide. All I've added here are two
23 other components that are important in our discussion.

24 The development portion of this slide is the same as
25 we saw before. The execution platform illustration shows both

1 a component of the virtual machine called "dexopt" -- and Andy
2 McFadden talked about that -- and a bytecode interpreter, which
3 he also discussed.

4 I have just a summary on the left that the dx tool
5 converts Java bytecode to the dex format. And then on the
6 computing device, such as a phone, dexopt loads these dex files
7 into the bytecode interpreter and optimizes them in various
8 ways.

9 The bytecode interpreter part of the virtual machine
10 executes bytecode instructions one by one. Those two
11 components are interrelated and refer to each other.

12 **Q.** So, now let's take a look at the patents. And let's start
13 with the '104 Patent.

14 This is Exhibit 4015. And can you walk us through
15 what the description is and the claims of the '104 Patent,
16 please?

17 **THE COURT:** Just to be clear, even though this is on
18 the screen, that particular patent, of course, will be in
19 evidence. Even though you're seeing it on the screen, this is
20 one document you will independently get anyway.

21 All right. Please go ahead.

22 **THE WITNESS:** Just to be brief about it, the title of
23 the patent is, "Method and Apparatus for Resolving Data
24 References in Generated Code."

25 It was invented by James Gosling, who is often

1 referred to as the father of Java. It's related to some other
2 patents. And this actual filing was -- was filed in 1999.

3 **BY MR. JACOBS:**

4 **Q.** Now, it refers to an earlier patent. Do you see that?

5 **A.** Yes, I do.

6 **Q.** And it says it's a reissue of another patent, the
7 5,367,685. Do you see that?

8 **A.** Yes.

9 **Q.** When was the 5,367,685 applied for?

10 **A.** That was filed in 1992, and issued November 22nd, 1994.

11 **Q.** Can you describe what the problem is that the '104 Patent
12 was designed to address?

13 **A.** Yes. Prior to this patent, there were two main ways of
14 compiling an executed code. So I'll talk about each one in
15 turn.

16 The first has to do with traditional compiled code.
17 This is scenario associated with the C programming language,
18 for example.

19 And in traditional compiled code such as C, a single
20 source code program has to be compiled differently to execute
21 on different machines.

22 So if you had a C program and you wanted to execute
23 it on -- well, here, a Sun architecture or Intel architecture,
24 you would use different compilers. Each would produce
25 machine-specific code for that architecture. And then those

1 programs would execute correctly on the particular architecture
2 they're compiled for, but they wouldn't be portable across
3 platforms from one to the other.

4 **MR. JACOBS:** Mr. Lee, can you put up TX 4015, the
5 '104 Patent. And can you go to column 1, lines 25 to 29.

6 (Document displayed.)

7 **BY MR. JACOBS:**

8 **Q.** So referring to the patent, Professor Mitchell, how does
9 what you just described relate to what's set forth here?

10 **A.** I believe that's what I tried to describe. The compiled
11 programming language, C as an example, a program called a
12 compiler compiles the source code and generates executable code
13 for specific computer architecture.

14 **Q.** And then if we go to lines 29 to 31.

15 **A.** This is an illustration, an explanation of one of the
16 reasons why the generated machine-specific code is not portable
17 or is specific to the layout on a particular computer. That's
18 illustrated in one of the figures that's -- that I have in the
19 slides.

20 **Q.** We'll get to that in a minute.

21 What are the advantages of compilation?

22 **A.** Uhm --

23 **Q.** Generally. You said this a minute ago, but what are the
24 advantages of the compiler approach?

25 **A.** Well, the compiler approach, because it's architecture

1 specific, can understand the memory layout and other factors.
2 And so you can produce efficient code. It's more efficient
3 than the interpreted scenario we'll look at in a minute.

4 **MR. JACOBS:** Mr. Lee, can you scroll down in that
5 column.

6 **BY MR. JACOBS:**

7 **Q.** So, if we go to line 44. So what are some of the problems
8 of the compiled approach?

9 **A.** One problem is portability. Another issue is that when
10 one portion of a program is changed, then the other portions of
11 the program have to be recompiled.

12 **Q.** And it refers there to "object-oriented programming." Do
13 you see that?

14 **A.** Yes.

15 **Q.** And Java is an object-oriented programming language,
16 right?

17 **A.** Yes, that's correct.

18 **Q.** So let's see if this is illustrated in the patent. What
19 is Fig. 1A showing us?

20 **A.** Fig. 1A depicts the situation when a sequence of
21 instructions compiled from source code refer to, in this case,
22 a portion of a data object.

23 So because the compiler in the compiled scenario
24 knows the layout -- can know the layout of data objects, it can
25 use a numeric reference that allows execution to proceed

1 directly to the second slot in the data object, without any
2 search.

3 So this is a fast process. It's dependent on things
4 known by the compiler in an architecture-specific scenario.
5 And if something was recompiled to change the data object
6 layout, you would have to recompile the program accessing it.

7 **Q.** So we talked about this a little bit with Mr. McFadden. I
8 talked with him about knowing the memory layout.

9 In the compiled approach, does the compiler need to
10 know the memory -- need to have information about the ultimate
11 memory layout?

12 **A.** Yes. There are particular ways that C and C++ programs
13 are structured, and particular kinds of information provided to
14 the compiler to allow that to work.

15 **Q.** Now, back to our drawing, then, of the Java world, would
16 the compiled approach work when you're aiming your program to
17 work on a whole bunch of different kinds of computers?

18 **A.** Uhm, no, because these memory layouts, this example and
19 others, are not known by the compiler.

20 **Q.** So, now let's turn to the interpreted approach which the
21 patent refers to. Can you explain what the interpreted
22 approach does?

23 **A.** So in an interpreted code, the interpreter steps through
24 the program incrementally and executes each instruction one at
25 a time.

1 So there's a little circle here showing that for code
2 derived from the source code the interpreter will fetch an
3 instruction, figure out what instruction it is, and then
4 execute it, and then repeat to go on to the next instruction.

5 So this involves reading a representation of source
6 code. And, typically, that involves symbolic references, which
7 are resolved more slowly. And, for that reason and others,
8 interpreted code is less efficient, execution is slower than in
9 the compiled scenario.

10 **Q.** So let's go back to the patent and look at Column 1, lines
11 58 to 64. So this reads:

12 "In an interpreted language, a computer
13 program called a translator translates the
14 source statements of a program into some
15 intermediate form, typically independent of
16 any computer instruction set."

17 What's that language about being independent
18 referring to in the testimony you gave?

19 **A.** The form that's used by the interpreter doesn't depend on
20 the computer that this program is going to be executed on.

21 **Q.** And then it says:

22 "References to data in the intermediate form
23 are not fully resolved before execution based
24 on the layout of the data objects that the
25 program deals with."

1 Do you see that?

2 **A.** Yes.

3 **Q.** And then it says:

4 "Instead, references to data are made on a
5 symbolic basis."

6 What's that saying in reference to the slide you were
7 just showing us about the --

8 **A.** When the interpreter reaches an instruction to access
9 data, the instruction may contain a symbolic reference. So the
10 symbolic reference gives the name of the data, but not where
11 it's located. So the interpreter has to go through a process
12 in order to find the data.

13 **Q.** And is that illustrated in the patent?

14 **A.** Yes. This is -- Fig. 1B of the patent shows an
15 instruction sequence, something based on the source code, a
16 representation in the left column of instructions.

17 This particular load instruction, "load" generally
18 means find some data and put it someplace where the computer
19 can operate on it next.

20 "Load" by a symbolic reference "y" is referring to
21 the name of the data. And the symbolic reference resolution
22 process is used to find where that data actually sits in memory
23 so that the load can be executed and the data can be moved
24 wherever the instruction calls for it to be moved.

25 **Q.** What did the '104 Patent do that was different?

1 **A.** The '104 Patent basically lets you have your cake and eat
2 it too. It combines the advantages of both of these. You can
3 have instructions that contain symbolic references, so those
4 are flexible and portable and architecture independent. But by
5 combining symbolic reference resolution and storing and later
6 retrieving the result, you can have the efficiency over time,
7 over the execution of a program that's normally associated with
8 compiled code.

9 So you have the flexibility of interpreted, but the
10 efficiency of compiled.

11 **Q.** So back in the day when this was written, there were some
12 diagrams in the patent that illustrate this hybrid approach?

13 **A.** Yeah. This is based on Fig. 8 of the patent.

14 Initially, the execution scenario associated and
15 described in this portion of the patent has a symbolic
16 reference. So the instruction says "load y by name." And then
17 that name is resolved to find out that "y" is "slot 2" in the
18 data object.

19 Once that's done, then the symbolic reference "y" is
20 no longer needed. We can replace "y" with the number "2." And
21 the next time this instruction is executed, the execution
22 platform can go directly to slot number 2.

23 Often, instructions are in things like loops in a
24 computer program. So the same instruction might get executed
25 many, many times if the code is in a loop that goes through all

1 of the elements of a list or any other kind of data structure.

2 **Q.** So you know that in my opening statement I tried to do --
3 draw analogy to Courtroom 8 and the third door on the left.

4 Can you, with maybe a little greater computer science
5 elegance, apply the discussion you have just given to that
6 analogy?

7 **A.** So in those terms, initially, the symbolic reference might
8 say, you know, get something from Courtroom 8, or see what's in
9 this courtroom, and load some data from it.

10 Then there's a directory for the building and some
11 instructions, if you look around in the hallway, on how to find
12 Courtroom 8. That's like the symbolic resolution process. And
13 then once you know that, every day when you come in you can
14 just come up the elevator and go to the left and come to the
15 door directly.

16 **Q.** And analogize that looping process that you just described
17 to the courtroom analogy.

18 **A.** Well, I know you've all been patient and coming here day
19 after day after day. So you're in a kind of computer loop of
20 coming back and resolving that reference over and over again.

21 **Q.** Did the '104 Patent, does it have any real-world
22 significance?

23 **A.** I mean, this is a, you know, huge, great idea in the
24 execution of Java. It allows you to have portable bytecode
25 that can run on a number of different architectures without

1 paying a significant performance penalty.

2 You get back the same performance over time that you
3 would get by compiling things in an architecture or
4 machine-specific way.

5 **Q.** Now, I'm looking now at the question of infringement. You
6 identified two ways in which Google infringes the '104 Patent?

7 **A.** Yes, that's correct.

8 **Q.** And can you briefly summarize those two, those two ways.

9 **A.** One of them is through the Resolve.c, I call it, the ways
10 that the bytecode interpreter runs and resolves symbolic
11 references, stores those symbolic references, and then uses
12 them -- I'm sorry, resolves the symbolic references to produce
13 numeric references, stores the numeric references, and then
14 uses them each time you go back to the same instruction.

15 And the other way that the '104 Patent is infringed
16 is through the dexopt portion of the virtual machine that
17 transforms instructions.

18 Andy McFadden talked about the IGET instructions.
19 Those are replaced by IGET_Quick instructions. And he also
20 described the way that works.

21 The IGET instructions have symbolic references where
22 they're rewritten by the quick instructions. The quick
23 instructions are quick because they have numeric reference
24 instead.

25 So there's a symbolic resolution process, yields

1 numeric references. Those numeric references are used to
2 rewrite the code to run more quickly afterwards.

3 Q. So let's turn to one of the claim charts that Judge Alsup
4 has described to us in this court.

5 This is for Claim 11 of the '104 Patent. And you are
6 looking at Claim 11 against Resolve.c or dexopt, or both?

7 A. Uhm, we'll be looking at it against Resolve.c.

8 Q. And just remind us, Resolve.c is?

9 A. The way that the bytecode interpreter resolves symbolic
10 references to find numeric references, stores them, and uses
11 them again and again.

12 Q. Now, you're aware that Google is challenging your
13 infringement analysis on the basis of this language that the
14 instructions must contain one or more symbolic references,
15 correct?

16 A. Yes.

17 Q. And what do you understand them to be arguing?

18 A. Uhm, I mean, it's -- it's hard to know exactly what they
19 would say after Andy McFadden's description, but somehow what
20 I've understood or inferred is that their position is that
21 indexes into a constant pool that are used in symbolic
22 reference resolution are not symbolic references.

23 Q. Now, did you, in doing your analysis, look at the Court's
24 definition of symbolic reference?

25 A. Yes, I did. This is a great, clear statement from the

1 Court about what symbolic reference means.

2 **Q.** Let's just walk through the language there.

3 Can you read it for us, so we can all make sure we
4 are all understanding what you are analyzing?

5 **A.** Sure. A symbolic reference is a reference, something that
6 refers to something, a reference that identifies data by a name
7 other than the numeric memory location of the data, and that is
8 resolved dynamically rather than statically.

9 So a reference is a reference to something. And this
10 says that a symbolic reference identifies that thing, the data
11 you want, by a name other than the numeric memory location that
12 that particular data you're looking for sits in.

13 And then we can see when a symbolic reference is
14 present, by looking, also, at the resolution process that
15 converts or produces a numeric memory reference from a symbolic
16 reference.

17 **Q.** And I want to ask you to focus on the language at the end,
18 just before the comma, "Other than the numeric memory location
19 of the data."

20 What was Mr. McFadden arguing about why the field
21 indexes, et cetera, were -- were not symbolic references, with
22 references to "of the data"?

23 **A.** I think it's --

24 **MR. VAN NEST:** Objection, Your Honor. Calls for
25 speculation. He can give his opinion, but characterizing what

1 Mr. McFadden is or is not testifying to is inappropriate.

2 **MR. JACOBS:** I'll take his version of the question,
3 Your Honor.

4 **THE COURT:** All right.

5 **THE WITNESS:** I'm sorry, what question should I
6 answer?

7 **BY MR. JACOBS:**

8 **Q.** What opinion do you have on Mr. McFadden's
9 characterization of the various Idxs as not symbolic
10 references, with the focus "of the data"?

11 **A.** I was here for a while through his discussion, and I
12 thought in some way he acknowledged that the index really is a
13 symbolic reference, according to the way I read this, because
14 it leads to -- it provides a name other than the memory --
15 numeric memory location that's resolved dynamically, and it
16 produces a numeric reference.

17 I think he was somehow saying, at some point, that an
18 index, because it initially is a number, is not a symbolic
19 reference. I didn't really understand the logic or sense of
20 that in this context.

21 **Q.** And he described the "fun" in his slide as a constant. Do
22 you recall that?

23 **A.** Yes.

24 **Q.** But he acknowledged that the constant was not the data
25 itself. Do you recall that?

1 A. Yes.

2 Q. And how does that relate, then, to this definition of
3 symbolic reference?

4 A. In a program, a constant is a name used by a programmer to
5 refer to a data value. So I don't know how much fun he had in
6 mind. We didn't see the source code.

7 But let's say "fun" was a name for the number 5.
8 Then the program using the word "fun" somewhere has to figure
9 out that the programmer meant the number 5, in this case. So
10 you would have to chase through the indices to find the name.
11 And then the data you want is the number 5, the constant value.
12 So that name in resolution is used to find the actual data
13 value 5 associated with it.

14 Q. And is "fun" that data value?

15 A. No. "Fun" is a name used by the programmer for a
16 different data value.

17 Q. Okay. Let's look at some of the evidence you examined.
18 You looked at Mr. Bornstein's description of dexopt?

19 A. Yeah. This is a simple statement that clearly says that a
20 dex file, as it arrives on a phone or other Android device, has
21 symbolic references to methods and fields.

22 Q. And then what does it describe as happening in the --
23 after it arrives on the device?

24 A. The rest of the paragraph describes reference resolution,
25 resolving the symbolic reference. So afterwards, after

1 resolution, it might be a simple integer vtable offset.

2 So integer vtable offset is a pretty technical term.
3 "vtable" stands for virtual function table. It's like the
4 figure in the patent where we saw slots.

5 So vtable offset would be a number like 2, meaning
6 the second slot in the vtable.

7 So he's referring here to resolving the symbolic
8 references to get numeric references, such as a vtable offset
9 or slot number. Then he also describes how that number can be
10 used instead of a string-based lookup.

11 So without using the original symbolic reference, you
12 can use that index into a vtable as the instruction is executed
13 to access the data that the program refers to.

14 **Q.** So in terms of -- so Mr. McFadden testified that this was
15 an accurate description. What does that say to you about
16 whether dexopt infringes?

17 **A.** Uhm, this is, in a nutshell, the core of the reason and
18 explanation of how dexopt infringes. We have symbolic
19 references. They are resolved to get numeric references. And
20 the whole purpose of this is to be able to use them
21 subsequently in execution. And he also covers this in this
22 very short paragraph.

23 **Q.** How about in the code, did you see indications in the code
24 that are consistent with your analysis?

25 **A.** Well, here's just one example. This is a comment that

1 talks about a portion of the code, and the fact that Android
2 converts symbolic references into pointers, which can be
3 numeric references.

4 **Q.** So looking at Claim 11, let's take the first element of
5 that:

6 "A memory containing intermediate form object
7 code constituted by a set of instructions,
8 certain of said instruction containing one or
9 more symbolic references."

10 Do you see that?

11 **A.** Yes.

12 **Q.** And, now, Google didn't underline the first part of this
13 but just let's walk the jury through how to understand this
14 claim and apply it to an Android device.

15 **A.** This is kind of a setup here that this applies to an
16 apparatus such as a computing device with a memory and a
17 processor.

18 So we know that an Android phone piece of hardware
19 has a memory. Things are stored on it, in a processor, so it
20 executes.

21 And, now, Android applications, things come in in dex
22 bytecode. We know that dex bytecode is -- provides a set of
23 instructions. And some of those instructions, as Dan Bornstein
24 just said in the past, we looked at, contain symbolic
25 references.

1 That's the first part of this.

2 **Q.** That's fine. Thank you.

3 And then did you see in "Resolve.c" further
4 explication of the resolution process that we're going to see
5 later on in the claim?

6 **A.** Uhm, yes. This -- this slide shows just some -- something
7 at the beginning of Resolve.c. It's a code file with a large
8 number of lines.

9 But this description says that this code in this
10 source code file, a portion of the Dalvik Virtual Machine,
11 resolves classes, methods, fields and strings.

12 And there are other parts of this description that
13 refer to resolving methods, fields, and so on.

14 So at the beginning, we can just see, reading this
15 file, that the developers here understood and explained that
16 this portion of the Dalvik Virtual Machine resolves symbolic
17 references to at least four kinds of things.

18 **Q.** And this is Trial Exhibit 47.6, for the note takers.

19 And then what is -- later on in 47.6, at page 8 to 9,
20 what is this code showing us?

21 **A.** So this is, you know, almost 400 lines down in the long
22 code file, or 450 lines down.

23 These are a few excerpts here, from the portion of
24 the code that has to do with resolving an instance field
25 reference.

1 So that an example of an instance field reference is
2 my test1 field in the code that I showed you. An instance
3 field means a field, a data field within an object that's an
4 instance of a class.

5 **Q.** Is ifieldIdx a symbolic reference or a numeric reference?

6 **A.** This is a symbolic reference. This is one of the indices
7 into the constant pool that Andy McFadden described. This
8 leads to the characters that are the symbolic name used by the
9 programmer for that field.

10 That's an input to the dvmResolveInstField method.
11 The name here is pretty descriptive. "Dvm" means Dalvik
12 Virtual Machine. "ResolveInstField" is just the abbreviation
13 for "resolve an instance field," as the comment above says more
14 clearly.

15 **Q.** So what does the body of this code do?

16 **A.** There's an excerpt here. Well, you can see from the
17 method declaration that it returns a pointer to an instance
18 field. And if you look further into the code -- I think the
19 next slide highlights -- we may or may not have it highlighted
20 separately.

21 Anyway, lines 419 to 421 show how the return value of
22 this resField, the actual resolved field numeric reference is
23 computed by calling another function with arguments that are
24 the name of the field and some other information.

25 **Q.** Now, let's look at the next portion of code, which is at

1 pages 1 to 2 of 47.6. What is this showing us? Maybe I
2 skipped one. Do you want to do slide 24?

3 **A.** You could ask the questions. What would you like?

4 **Q.** Let's go to the next -- let's go to classes, methods, and
5 strings resolved the same way, on slide 25.

6 **A.** Okay. So we've seen some of the code excerpts that show
7 how field names are resolved. This is another portion of the
8 code that has to do with resolving classes. These are done
9 similarly. The comment, it describes that. And if you look
10 below, there's a function called very much parallel to the
11 field case.

12 The comment here says, "Find the class corresponding
13 to the classIdx." Then it explains the classIdx maps to a
14 class name string. And then that's used in this resolution
15 process.

16 So this is, again, showing a class name, a symbolic
17 reference being used and resolved to a numeric reference.

18 **Q.** Okay. So let's review where we are in Claim 11.

19 **A.** Okay. So we've talked about the memory and processor and
20 the instructions that contain symbolic references. And I've
21 shown you sections of the code for resolving or determining a
22 numeric reference corresponding to a symbolic reference both
23 for the example of fields -- went through that in more
24 detail -- and for the example of classes. For methods and
25 other things that are resolved, there are similar sections of

1 code that do that in the same way.

2 **Q.** So, now, let's go on to the next limitation. I think the
3 jury understands that each of these clauses are limitations.

4 So let's take a look at storing the numerical
5 references. And we're going to go back to "Resolve.c" at 47
6 point -- Trial Exhibit 47.6, at pages 8 to 9. What do we see?

7 **A.** So this is, again, a code excerpt. The whole point of
8 getting a numeric reference is store it and use it over and
9 again.

10 Here is the function that sets the resolved field.
11 So that's DvmDex. That's Dalvik Virtual Machine for processing
12 dex code. Set the resolved field. And it has as an argument.
13 The last one is this resolved field. So that's the thing that
14 takes the resolved field and takes it somewhere.

15 The comment above it is nice and descriptive. It
16 says, "Add something to our data structure so we don't have to
17 jump through hoops." That means go through the whole
18 resolution process again.

19 This is where you get the speed up, is you store
20 something that was relatively computationally expensive to
21 find, so that next time you come to this instruction you don't
22 have to do that again.

23 **Q.** So that was Element C. Now let's look at Element D, which
24 is, "Obtaining data in accordance to said numerical
25 references." Can you explain how Android does this?

1 **A.** Yeah. So this is a different kind of code, for anyone
2 who's trying to make sense of it.

3 This is assembly code. This is the -- a portion of
4 the bytecode interpreter that loops through the instructions
5 one at a time and executes them. This is even harder to read
6 than other kinds of computer code.

7 The important points here are that, as line 1970
8 shows, this is for executing the OP_IGET, the field -- the
9 access get data for a field instruction.

10 And in this there's a line that refers to this
11 resField or resolved field. And it tests whether the field
12 "resolved field" is actually there. "No" would mean it's not
13 there, it hasn't been resolved. If the resolved entry is not
14 null, then it's used.

15 So this is showing how the most detailed part of
16 this, in some sense, the execution of these instructions, tests
17 to see if symbolic reference has already been resolved. And if
18 they have, use the stored numeric reference to speed up the
19 execution of the system.

20 **Q.** So with the focus on this slide, which is TX 47.13 at page
21 34, line 1970 to line 1993, how does this map to the find
22 Courtroom 8 analogy?

23 **A.** So this is -- if you come in in the morning, and you
24 remember where courtroom 8 is, you just go right there. And if
25 you forget, well, you can always look at the building directory

1 and find it in the more complicated way.

2 **Q.** So, to summarize, have we walked through each of the
3 limitations of Claim 11 of the '104 Patent?

4 **A.** Yes. One at a time, we've covered all four parts of the
5 claim.

6 **Q.** I think, just to be sure, we didn't explicitly talk about,
7 "A processor configured to execute said instructions containing
8 one or more symbolic references." What is that referring to?

9 **A.** So, I mentioned an Android device having a processor and a
10 memory. Once Android is loaded and some application or even
11 the system to start running, then "processor configured to
12 execute instructions" means a processor with the Dalvik Virtual
13 Machine and runtime environment loaded.

14 **THE COURT:** So are you about to go to a new claim?

15 **MR. JACOBS:** Yes.

16 **THE COURT:** This would be a great time to take a
17 15-minute break. Please remember the admonition.

18 **THE CLERK:** All rise.

19 (Jury out at 11:42 a.m.)

20 **THE COURT:** All right. Please be seated.

21 And, Professor Mitchell, you can take a 15-minute
22 break, as well.

23 **THE WITNESS:** Thank you.

24 **THE COURT:** I have a question for you on the SSO part
25 of the case.

1 Did we have any testimony that -- that compared
2 particular classes, Android versus Java, to determine whether
3 the methods were listed in the same order, same sequence?

4 **MR. JACOBS:** Yes, I believe so, Your Honor. We had
5 it in the general case and, I think, also in the specific case.

6 **THE COURT:** So who testified to that?

7 **MR. JACOBS:** Well, what I was recalling was the
8 admissions I listed on Dr. Astrachan, in which he
9 acknowledged -- I actually think I have for this afternoon --

10 **THE COURT:** Well, let me just pose the question, and
11 maybe in the afternoon session you can all address that.

12 But the question I'm posing is, if we were to take,
13 say, ten random classes, and within each class there would be
14 some -- I don't know how many, a number of methods, would we
15 find that if we did the listing, that the methods show up in
16 exactly the same order?

17 **MR. JACOBS:** I understand the question more precisely
18 now, Your Honor. We'll hunt that down.

19 **THE COURT:** All right. Now, anything you need me for
20 before we take our break?

21 **MR. JACOBS:** Not from us, Your Honor.

22 **MR. VAN NEST:** No, Your Honor.

23 **THE COURT:** At this rate, we're not going to finish
24 today. We only did one claim.

25 **MR. JACOBS:** The others will be quicker, Your Honor,

1 because the issue is largely the same.

2 **THE COURT:** All right. See you in a few minutes.

3 (Whereupon there was a recess in the proceedings
4 from 11:44 until 11:58 a.m.)

5 **THE COURT:** Back to work. Any issues for the judge?

6 **MR. JACOBS:** No, your Honor.

7 **MR. VAN NEST:** No, your Honor.

8 **THE COURT:** All right. The witness can have a seat
9 and we'll bring in the jury.

10 (Jury enters the courtroom at 12:00 p.m.)

11 **THE COURT:** Welcome. Be seated, please. Ready?

12 All right. Mr. Jacobs, the floor is yours.

13 **MR. JACOBS:** Thank you, your Honor.

14 Can we have slide -- yes.

15 (Document displayed)

16 **BY MR. JACOBS:**

17 **Q.** So this slide, Android's Dalvik VM infringes Claim 39.

18 What can you tell us about Claim 39, Professor
19 Mitchell?

20 **A.** Claim 39 we see is very similar. This, to begin with,
21 involves a computer-implemented method. So that's a method
22 that's implemented in our case in software to run on a
23 computer, and I can start to walk through that.

24 **Q.** When you said a "computer," a computer-implemented method,
25 does that method get practiced by a phone running Android?

1 A. Yes.

2 Q. And so the word "computer" there, how does that apply to
3 Android, to an Android phone?

4 A. An Android phone has a computer processor and other
5 computer components to execute computer software in the normal
6 way that other computers work.

7 Q. Now, if we look at the Google underlining in this claim,
8 we see "a symbolic field reference" underlined and then some
9 language at the lower part of the claim; do you see that?

10 A. Yes.

11 Q. So do you believe -- is it your opinion that an Android
12 phone infringes Claim 39?

13 A. Yes.

14 Q. And can you explain to us the basis for that opinion?

15 A. It involves the entire claim. To focus on the underlined
16 part first, we already discussed symbolic references, symbolic
17 field references. Our example, symbolic references that are
18 present in the Dalvik bytecode.

19 And then "data from a storage location identified by
20 a numeric reference," we saw in the interpreter, the assembly
21 code, where data from a storage location identified by a
22 numeric reference is accessed and used and thereafter used for
23 operation. And, "when the instruction contains a symbolic
24 field reference," then the symbolic field reference is resolved
25 to get that numeric reference that's used.

1 So the order of the claim is a little bit backwards
2 from the order that we went through in Claim 11. It's a little
3 bit confusing, but the parts of this are the same and infringed
4 in the same way for the same reason.

5 **Q.** Now, there is a word that's used in Claim 39 that wasn't
6 used in Claim 11 and it's highlighted here on the next slide.
7 You can see it on the screen perhaps.

8 **A.** Yes. So our discussion used the word "resolve," but
9 Claim 11 doesn't actually contain that word.

10 **Q.** And do you remember what Claim 11 said instead of
11 "resolve"? I think it was "determine," right?

12 **A.** So determining a numeric reference corresponding to a
13 symbolic reference.

14 **Q.** And here in Claim 39 the word -- what word is being used?

15 **A.** The word "resolved" is used.

16 **Q.** Now, did you apply a definition from the Court on
17 "resolve" and "resolving" in your analysis?

18 **A.** Yes. So the Court claim construction ruling gives
19 specific meanings to "resolve" and "symbolic reference" and
20 other terms. So I used the Court's construction for "resolve"
21 and "resolving" in understanding this claim and comparing it to
22 the Android software and system.

23 **Q.** And that interpretation is:

24 "At least determining the numerical
25 memory-location reference that corresponds to

1 the symbolic reference."

2 Correct?

3 **A.** That's correct.

4 **Q.** Now, can you show us in the code where you see resolving
5 occur?

6 **A.** We went through this before, but just to repeat it, to be
7 clear about this particular claim. One example is in the
8 Resolve.c source code file that's part -- defines part of the
9 Dalvik VM.

10 Here are some excerpts describing resolving an
11 instance field reference and showing a function call that's
12 used in that process. This particular resolve instance field
13 function call takes a symbolic reference as an argument. This
14 ifieldIdx is a symbolic reference that comes from the
15 instruction, and then this function returns the resolved field
16 that's produced by the resolution process.

17 **Q.** So does Resolve.c determine or resolve the numerical
18 reference corresponding to a symbolic reference?

19 **A.** Yes. And that shown directly in the source code excerpts.

20 **Q.** Briefly, 419.

21 **A.** There is the resolved field and it's produced by another
22 call to a DVM function that takes a field name and a type name
23 as arguments. So that particular function call determines the
24 memory location -- numerical memory location reference
25 corresponding to the symbolic reference that's passed as a

1 argument to that method. So that meets exactly what the claim
2 construction is for resolve.

3 Q. Then there is the phrase in Claim 39 that we hadn't seen
4 in Claim 11, "thereafter used." So how did you analyze that?

5 A. In this computing context the "thereafter" is the phase of
6 execution after the symbolic reference is resolved. We saw
7 source code that stores the symbolic reference and here in the
8 next slide there is a -- again, the assembly code for the
9 interpreter that "thereafter uses" the numeric references
10 produced by resolving symbolic references.

11 Q. So during execution does the Dalvik VM thereafter use
12 stored numeric references?

13 A. Yes.

14 Q. Let's take a look at claim 40. What are the differences
15 between Claim 40 and Claim 39?

16 A. One difference is that Claim 39 refers to a
17 "computer-implemented method." Claim 40 talks about a "data
18 processing system," which has specific parts listed in the
19 claim; such as a processor and a memory. These are similar to
20 the processor and memory portions of Claim 11.

21 So we have already talked about how an Android device
22 with software loaded has a processor and a memory and the
23 memory does the things described in the Android source code.

24 Q. Are there any -- are there any disagreements underlined by
25 Google in Claim 40 as compared with Claim 39 -- that are

1 different from the underlining in Claim 39?

2 **A.** I think the underlined parts are exactly the same. So
3 that means the points that have been -- where there has been
4 some disagreement are identical for these two claims.

5 **Q.** Now, in Claim 40 it says:

6 "Analyze each instruction to determine
7 whether it contains a symbolic field
8 reference."

9 And then:

10 "Execute the program by performing an
11 operation identified by each instruction,
12 wherein data from a storage location
13 identified by a numeric reference is
14 thereafter used for the operation when the
15 instruction contains a symbolic field
16 reference."

17 **A.** Yes.

18 **Q.** So can you just walk us through very crisply how Claim 40
19 is infringed by Dalvik with reference to those underlined
20 items?

21 **A.** Okay. First of all, the Dalvik bytecode has symbolic
22 field references, as we've discussed.

23 And then "data from a storage location identified by
24 a numeric reference," that includes the data value associated
25 with a field, identified by numeric reference the exact offset

1 of that field in the object as it's manipulated by the program.
2 That numeric reference is used after resolution when the
3 instruction originally contained a symbolic field reference and
4 that field reference was resolved using Resolve.c and the other
5 source code functions that we discussed and looked at.

6 **Q.** So is your analysis of Claim 40 any different from your
7 analysis of Claim 39?

8 **A.** No. Only just to note the processor memory differences.

9 **Q.** So now let's go to Claim 27.

10 **A.** I have 41, if you'd like.

11 **Q.** And the highlighting on Claim 27. So Claim 27 you're
12 going to analyze against dexopt, is that correct?

13 I'm sorry, I skipped. Let's do Claim 41, please.

14 **A.** 41 and 39 are the same beyond the first through lines of
15 the claim.

16 **Q.** And the first few lines in 41 read:

17 "A computer program product containing
18 control instructions for causing a computer
19 to perform a method."

20 What's that driving at?

21 **A.** So a product -- an example would be an Android phone that
22 contains instructions for causing the computer, the processor
23 in the phone, to perform a method. The method is described in
24 software. And we've already seen that this Android software
25 meets all of the limitations of this claim.

1 Q. Thank you.

2 And so are there any other analytical differences --
3 are there any analytical differences in terms of the
4 underlining as between Claim 41 and Claim 39?

5 A. No. The claims are the same past the first few lines.
6 The way you understand it and see that it's met by Android is
7 exactly the same.

8 Q. So does Android's Dalvik Virtual Machine infringe
9 Claim 49 [sic], insofar as the underlining is concerned,
10 in the same way that it infringes Claim 39?

11 A. Yes. 41.

12 Q. Okay. Now, let's do Claim 27. And you analyzed Claim 27
13 not against the bytecode interpreter for purposes of this
14 trial, but for -- but against dexopt, correct?

15 A. That's correct.

16 Q. So how does dexopt infringe Claim 27?

17 A. Claim 27 describes or calls for "generating a set of new
18 instructions for the program."

19 The new instructions contain numeric references, and
20 those numeric references in the new instructions result from a
21 routine that resolves some data -- symbolic data references in
22 the original instructions.

23 Q. Now, this claim doesn't contain the word "contain," does
24 it, Dr. Mitchell?

25 A. No.

1 Q. It just says:

2 "...resolve any symbolic data references in
3 the set of original instructions."

4 Do you see that?

5 A. Yes.

6 Q. Okay. So let's walk through your analysis of Claim 27 and
7 bring us back with -- to the diagram on slide 42 of the Android
8 Platform components.

9 (Document displayed)

10 A. Yes. So this slide is just to remind us of where these
11 components sit in an Android system. The top half of this is
12 the developer environment and the bottom half is the Dalvik
13 runtime, the installed runtime system that's on an Android
14 phone used by a phone user or used by a developer who's testing
15 a phone or building the system in another way.

16 Dexopt and the bytecode interpreter both are part of
17 this Dalvik runtime. They are interoperable. They call -- the
18 dexopt calls functions that are defined in the end part of the
19 bytecode interpreter. They are really closely linked in that
20 way.

21 Q. Now, did you look in the documentation to determine
22 whether dexopt generates new -- actually, let's go back and
23 just focus on the highlighting for a minute.

24 Google underlined the "generating a set of new
25 instructions language."

1 **A.** Yes. That's -- that's the part that I talked through in
2 explaining how this is infringed.

3 **Q.** So, now let's go to the dexopt documentation and how did
4 you -- what did you see here that relates to whether the
5 "generate a new instructions language" of Claim 27 is met by an
6 Android device?

7 **A.** Okay. This documentation is a little bit easier to read
8 than source code. And it talks about what the Dalvik optimizer
9 or dexopt component does.

10 So the first bullet here says that:

11 "For virtual method calls, replace the method
12 index with a vtable index."

13 So that corresponds to the language that Dan
14 Bornstein used in talking about symbolic references and
15 resolving them and producing a vtable index as a numeric
16 reference.

17 The second bullet talks about instance fields.
18 That's the example I used with the iget instruction in the
19 Dalvik bytecode. And the Dalvik optimizer or dexopt takes an
20 instance field get or put instruction and replaces the field
21 index -- the thing that we've seen is a symbolic reference --
22 with a byte offset, something that is a numeric reference.

23 **Q.** Let's just repeat that last point, please.

24 What is the symbolic reference and what is the
25 numeric reference in this second bullet?

1 **A.** The field index that's in the original iget instruction is
2 a symbolic reference with a new instruction. An iget quick
3 instruction is created by dexopt. That new instruction was a
4 byte offset, which is a numeric reference just like the number
5 two in slot two in the patent illustration.

6 **Q.** Did you look at the -- at the code and the comments in the
7 code to further analyze this question?

8 **A.** Yeah. I read a lot of code, including the comments and
9 the instructions below the comments.

10 Here is one comment that explains how this works and
11 talks about these indices, indexes, and how they are resolved
12 and replaced with a byte offset, such as the number two in slot
13 two.

14 So if you want I'll just read it again. The format
15 of this instruction has something called field@CCCC. CCCC
16 indicates an index in a certain format, and --

17 **Q.** And it's described here as the, CCCC is the --

18 **A.** Field reference constant pool offset. So field reference,
19 it's a reference. And constant pool offset means the index
20 into the constant pool that provides symbolic references for
21 these bytecode instructions.

22 **Q.** And then what happens? What does this comment describe as
23 what happens to those field references?

24 **A.** What dexopt tries to do, what we want to do in the code is
25 replace the OpCode, that's the bytecode instruction, with a new

1 one, a quickOpCode. And replace CCCC, replace the symbolic
2 reference with a bytecode offset from the start of the object.
3 So that's a numeric reference exactly like the number slot --
4 number two in slot two.

5 The byte offset from the start is -- if you number
6 the fields and the object, the offset is how many fields you
7 count down in order to find the field that contains the data
8 that the program refers to.

9 **Q.** What's the new instruction that's generated according to
10 this comment?

11 **A.** Here it's in quotes. It's just "quickOpc." We'll see on
12 another slide exactly how these are named.

13 The original instructions don't have the word "quick"
14 and new ones that contain numeric references have the word
15 "quick" in them because they are quicker. Because this numeric
16 reference is already produced, there is no need to resolve a
17 symbolic reference.

18 **Q.** So now we're looking at some more code. TX 47.2 at Page 4
19 we are look at Android Optimize.c.

20 **A.** This is a code that goes through and branches according to
21 the name of the instruction. So a "switch" instruction in c
22 code means jump to one of the cases, depending on the
23 instruction. So right below the "switch" instruction, there is
24 a case for OP_IGET. That's an instruction that Andy McFadden
25 described. This is an instruction that accesses the data in

1 the field of an object.

2 So for this case, all of those cases that's begin

3 OP_IGET lead to the code below that says there is a quickOpC

4 that's now called OP_IGET_QUICK, the quick version of IGET.

5 Then the few lines below that say how that is handled to effect

6 the exchange from one OpCode to another.

7 **Q.** How does this illustrate dexopt generating new quick

8 instructions containing numeric references?

9 **A.** This is the beginning of the process, but it's --

10 detecting an instruction that can be changed setting up the

11 data so that it will be changed to a quick form and then it

12 says "goto rewrite_inst_field."

13 So the next step will be to go rewrite the portion of

14 the instruction that has the reference to replace the symbolic

15 reference by a numeric reference.

16 **Q.** So let's now go to the next slide. This is showing 47.2

17 at Page 12, Lines 643 to 653 of the code.

18 What is this showing us?

19 **A.** Yes. This is another step along the way. Here is a call

20 to the ResolveInstField method that's part of the dexopt and

21 dvm software.

22 And it takes as an argument the fieldIdx, this index

23 into the constant pool that is a symbolic reference. And it

24 produces as an output the thing on the left of this line, the

25 instField that's then used to rewrite the instruction in the

1 next slide.

2 **Q.** So now we're looking at Lines 662 and 663.

3 **A.** These both begin with "updateCode." Those are calls to
4 functions or methods that change the bytecode and they change
5 it at certain instruction numbers using the quickOpc that we
6 saw in the previous code excerpt is set to be the name of the
7 new quick instruction.

8 Then you can see in the second line that's what's
9 being written into the instruction is the byteOffset. So
10 that's the numeric reference, like two in slot number two.

11 **Q.** And then turning to 47.13 at Line 5966 to 5982. What do
12 we see?

13 **A.** So this is just to complete the story on how that's used.
14 This is another case for the bytecode interpreter for this new
15 instruction.

16 So if you look at the top of the excerpt 5966, that's
17 the name of the instruction and the bytecode that gets you to
18 this part of the interpreter. Here it's called IGET_QUICK. So
19 the lines below that are the interpreter that executes a quick
20 instruction.

21 And you see down in about the middle of the code
22 excerpt there is a highlighted line that says "read the field
23 bite offset."

24 So in this quick instruction the interpreter is using
25 the numeric reference that was written into the bytecode by the

1 function calls on the previous slide.

2 **Q.** Let's go back to the definition of symbolic reference.

3 There is a discussion in the -- there is language in the

4 definition that is "resolved dynamically rather than

5 statistically." Do you see that?

6 **A.** Yes.

7 **Q.** And you will recall Mr. McFadden and I had a discussion

8 about when and whether -- when dexopt runs and what information

9 it depends on from the -- in the handset itself.

10 **A.** Right.

11 **Q.** And what did you understand Mr. McFadden's testimony to be

12 as it bears on the question of static versus dynamic?

13 **A.** From what I recall from what he said, is that dexopt

14 requires information that is only -- that depends on the

15 platform, the firmware, the software that's installed; things

16 that are only known in the runtime environment of the actual

17 phone.

18 **MR. VAN NEST:** Objection, your Honor. Move to

19 strike. That is not consistent with what the witness said.

20 And I object to him characterizing what Mr. McFadden said. He

21 should give his own opinions.

22 **THE COURT:** Well, no. It's okay for him to -- I

23 assume that's why you all have these experts in the courtroom,

24 so they can comment on what's been heard.

25 But I don't remember what Mr. McFadden said. It

1 would be amazing if the jury remembered. He was very hard to
2 follow.

3 So maybe the thing to do would be just quote it. Why
4 can't we get it out and quote it?

5 **MR. VAN NEST:** That would be a lot better than what
6 we're doing now.

7 **THE COURT:** Let's quote the -- do you have the
8 transcript ready? Is it easy to find?

9 **MR. JACOBS:** Give it a shot, your Honor.

10 **THE COURT:** For the time being, here is the way we're
11 going to get through this.

12 Just put an asterisk in your notes that there was a
13 vigorous objection made to the way in which this testimony was
14 characterized and that we may need to get a reedback of what
15 that actual testimony was to make sure the witness has not
16 mischaracterized what the witness actually said.

17 And as long as we have that caveat in there, I think
18 we'll just proceed. And by the time we can get to it, we'll
19 just read back to you slowly what it was that the witness said
20 so you can judge for yourself how close the witness on the
21 stand got to it.

22 Let's go ahead with the next question.

23 **BY MR. JACOBS:**

24 **Q.** What is your opinion on Mr. McFadden's testimony as it
25 bears on the question of runtime, usage of dexopt?

1 **A.** My opinion is that he explained and gave some reasons why
2 this operation done by dexopt requires runtime information, and
3 there are a couple of different reasons why that's true.

4 **Q.** What is your opinion as to what pieces of information
5 dexopt requires in order to operate that you -- that is
6 available from the runtime environment?

7 **A.** So one piece of information is the actual location of
8 things, such as classes in the runtime environment. The source
9 code doesn't contain any locations. It is only when the code
10 is loaded into a computer memory on the phone that the classes
11 and other things defined by the source code reside in a
12 particular place in memory. So you can't really have a numeric
13 reference to a class until that class is loaded into memory.

14 **Q.** That's one kind of information.

15 What other kinds of information were you just
16 referring to?

17 **A.** Well, the dexopt functions call the bytecode interpreter
18 functions and are interrelated. So it's the same software base
19 running in the same way.

20 There is also the architecture and some other things
21 that Mr. McFadden referred to that are important.

22 **Q.** Okay. Now, I also talked with Mr. McFadden about some
23 documentation for dexopt. And you looked at this documentation
24 as well, correct?

25 **A.** Yes.

1 Q. Let's look at TX 105 Pages 1 to 3.

2 What did this indicate to you about whether dexopt
3 needs runtime information?

4 A. This is further explanation along the lines I just stated.
5 This says desktop dexopt in the terms of this documentation is
6 a back door into the VM. That means it's another entry point,
7 another way of starting and getting into the runtime
8 environment of the virtual machine.

9 And it explains further that it starts the virtual
10 machine up, does this kind of dynamic boot process there, loads
11 files from the class path because, as I explained, we need to
12 know where they sit in memory in order to find numeric
13 references, and then sets about verifying and optimizing and
14 doing other things with that bytecode.

15 So this is the sense in which it's dynamic and it's
16 part of the runtime environment of the Android platform.

17 Q. So I think we've just been through Claim 27.

18 Now let's look at Claim 29. Same underlining it
19 looks like. Do you see that?

20 A. Yes. These claims are the same beyond the first few
21 lines. So the way -- the issue and limitation about generating
22 new instructions with numeric references in the new
23 instructions, those numeric references coming from resolving
24 symbolic references in original instructions, that part that's
25 been disputed and is going to be part of -- an important part

1 of this case is the same in both claims.

2 The difference really is at the top where Claim 29
3 talks about, "A computer program product containing
4 instructions for causing a computer to perform a method, the
5 method comprising these steps."

6 The Android source code is a product that contains
7 instructions for causing a computer to do specific things.
8 That's what programs are for.

9 And so that source code or the product produced by
10 compiling and building the system from that source code
11 contains the methods that that source code implements. And
12 I've walked through how that method works and why it meets the
13 limitations of this claim.

14 **Q.** So we walked through the claims of the '104 patent and
15 we've discussed your analysis of Android.

16 Do you have any reason to believe that third-party
17 OEMs who install Android on their phones modify the code that
18 you analyzed for purposes of your '104 infringement analysis?

19 **A.** No, I don't.

20 **Q.** And do you have reasons to believe they don't modify it?

21 **A.** This is an important performance, both of these features,
22 the dexopt and the Resolve.c. Symbolic reference resolution
23 and associated steps that I just walked through are important
24 parts of the performance of the system.

25 So it just doesn't make any sense for someone selling

1 Android phones to disable part of the software and make their
2 phone work more poorly. I don't see why anyone would want to
3 do that.

4 Q. In all of your review of evidence in this case did you
5 ever see any indications that such modifications were made by
6 OEMs?

7 A. No, I did not.

8 Q. Okay. Let's turn to the '520 patent.

9 (Document displayed)

10 Q. And just explain briefly that what we see here on the
11 screen from TX 4011, U.S. patent number 6,061,520.

12 A. This is the front page of the '520 patent and, among other
13 things, the front page tells us the inventors and the filing
14 date. The inventors are two people, Frank Yellin and Richard
15 Tuck. And this patent was filed on April 7th, 1998.

16 Q. Now, what's the problem that prompted the '520 patented
17 invention?

18 A. This patent and its invention addresses an issue with
19 static arrays and the way they are initialized.

20 So an array, as we've talked about, is the field IDs
21 and other things. Now, the constant pool is just a list of
22 things in order and the things in order are numbered zero, one,
23 two; or one, two, three. They are numbered so that you can
24 access things out of the list by giving the computer the number
25 of the entry you want from the array.

1 It's common for programs to have long arrays that
2 contain data and those arrays have to be initialized; that is,
3 the list has to be created with the data stored in order in the
4 list.

5 This slide shows a line of Java source code that
6 defines a static array and tells -- and lists the values to be
7 put in that list. So this static array called setup should
8 have four locations and the four locations in the list for data
9 values in the list are the numbers one, two, three, four.

10 Now, when that source code is compiled, the Java
11 compiler produces a fairly long and a little bit tedious
12 bytecode program for storing one in the zero location, two in
13 the location after that, three in the location after that, and
14 then finally four, and then being done with a process.

15 So you can see that although the source code is just
16 one line, the bytecode is long, takes up a lot more space than
17 the list of numbers and will take some time to execute, too,
18 because it stores these initial values in the array in a
19 somewhat laborious way.

20 **Q.** So in this slide the Java compiler creates many bytecode
21 instructions to initialize the static array. We're looking at
22 source code in the patent? That's a -- that statement "static
23 int setup" is from the patent itself?

24 **A.** I believe this is the example that's in the patent.

25 **Q.** And that's at Column 1, Line 65.

1 And then is the compiled Java bytecode the .class
2 file also in the patent?

3 **A.** Yes. This is directly from the patent's specification,
4 the problem that the inventors were trying to solve.

5 **Q.** And just -- maybe just do this once so we can get past it.
6 What is a clinit method?

7 **A.** This is for initializing a class. So this is just the
8 name in the bytecode for something that has to be run when the
9 class is loaded into the virtual machine. So that if any
10 program refers to the data in the array, the data is there and
11 accessible to the program.

12 **Q.** Is the -- sorry, one more question about this.

13 Is the clinit method, is it directed to static
14 arrays?

15 **A.** This example is for static arrays, yes.

16 **Q.** How about for other usages?

17 **A.** I don't recall. The things that are important for the
18 patent. This particular method is used for static array
19 initialization.

20 **Q.** And so clinit is -- when we've heard about methods in
21 Phase 1, clinit is the name of a method, is that right?

22 **A.** Yes.

23 **Q.** Okay. So what's the solution?

24 **A.** The solution is to go through and simulate what the
25 virtual machine would do with this bytecode and produce based

1 on that simulation some different kind of instruction that the
2 virtual machine can use to initialize the array more directly
3 and without reading as many instructions from the bytecode
4 program.

5 **Q.** So in my opening comments I analogized this to one kind of
6 grocery shopping, and this to another kind, and described what
7 I thought might -- analogize to the simulation.

8 Again, with a little more computer science expertise,
9 can you work through my analogy as it applies to what you've
10 shown here?

11 **A.** The bytecode compiler, the Java compiler is not very --
12 it's very literal and doesn't really understand the layout of
13 the grocery store or the specifics of that.

14 So if you have a list that says, "Go buy apples.
15 Then go buy bread. Then go buy tortillas," or whatever you
16 buy, then the bytecode program might tell you, "Go this place
17 and buy that thing, and then go back to the beginning of the
18 store, and go to the second place and buy the second thing."
19 So you'll see a repeated scene of instructions that look like
20 you're starting from the same place and walking down some path
21 over and over again.

22 The simplified shorter instruction that takes up less
23 space and is executed more quickly, just lists what has to be
24 done directly. So it can be done more efficiently by the
25 virtual machine.

1 Q. Let's look at Claim 1 of the '520 patent, and here the
2 underlining is about the simulating step, correct?

3 A. Yes.

4 Q. And we've heard in opening statement from Google that they
5 challenge whether Android simulates execution of the bytecodes.

6 So can you explain your view of this argument?

7 A. I'm sorry. Which argument?

8 Q. That Android does not simulate.

9 A. Well, I mean, I believe that it does. And I have the --
10 the reason for that is based on the source code, what the
11 source code says, how it works, what it achieves, and all of
12 those things.

13 Q. So let's walk through that. Let's look at the platform
14 components again.

15 And how does this platform components slide relate to
16 the '520 patents -- patent?

17 A. The important thing is that this occurs in the dx tool in
18 the top half in the developer portion of the platform.

19 The Android SDK, the software that Google provides
20 and recommends the developers use, has the standard Java
21 bytecode compiler from Oracle followed by this tool that
22 creates -- that converts Java bytecode to dex code and that's
23 the place where this patent invention is used to recognize by
24 simulation this complicated sequence of instructions and
25 replace them with something simpler that goes in the outgoing

1 bytecode.

2 **Q.** Did you see indications from Dan Bornstein presentations
3 that this static array initialization efficiency technique of
4 the '520 patent is used by the dx tool?

5 **A.** Yes. This slide shows two smaller images of slides that
6 Dan Bornstein used, and he described this as a feature that he
7 liked in the dx tool. And the slide on the left and the slide
8 on the right correspond to the figures in the patent in this
9 way.

10 The figure -- the slide on the left shows a bytecode
11 sequence used to initialize an array. And the slide on the
12 right shows what the dx tool produces, which has a single
13 instruction for initializing an array followed by the actual
14 data values to be put in the array. So that looks a lot
15 like -- I mean, it's a little more -- the dex form of it is
16 written differently, but it's the same idea as the previous
17 slide I showed you for where the patent says how you can
18 produce one instruction followed by data and use that to
19 initialize the array efficiently.

20 **Q.** And so that's looking at Slide 42 of Mr. Bornstein's
21 presentation.

22 Can you just comment briefly on slide 44 of his
23 presentation?

24 **A.** Yes. So 42 is the bytecode for initializing the array.

25 44 is the dex code that contains the simpler, more

1 concise single instruction and data to go into the array.

2 **Q.** So let's go back to this simulating step. Let's look at
3 the code that you looked at to determine whether the dx tool
4 simulates execution of Java bytecode.

5 And we're looking at Slide No. 61, but the exhibit,
6 for the note takers, is TX 46.16 at Lines 37 to 56.

7 What does this code tell us?

8 **A.** This is source code that defines the behavior of the dx
9 tool and it contains comments written by the engineers
10 indicating how they think about it and what this code actually
11 does.

12 So they say straight out: This is a class, the class
13 defined by the source code, that knows how to simulate the
14 effects of executing bytecode.

15 So to begin with, we know that the Google engineers
16 that wrote this code thought of this as simulation and thought
17 that was the best way to describe it.

18 Then the class that's defined here is called
19 simulator.

20 And then there is another instruction talking about
21 things to use in simulation.

22 So just to begin with, this is called simulation in
23 the Google source code.

24 **Q.** And what is the name of the file?

25 **A.** The file also has the name Simulator.java. The file name

1 and the class name also both indicate this is a simulator.

2 **Q.** So now we're look at lines 86 through 107 of TX 46.16.
3 What does this tell us?

4 **A.** This is from that same source code file. So this is a
5 portion of the simulator class. It's the actual method that
6 can be called to start the simulation. And this method is
7 called "simulate."

8 So that means a programmer asking for these steps to
9 be executed calls and writes the instruction. You know, the
10 method calls simulate and then give some instructions for this
11 to apply to.

12 At the top this comment also says:

13 "This method simulates the effect of
14 executing the given basic block."

15 Namely, for the case of the clinit method, simulate
16 the effect of executing the instructions in that.

17 **Q.** Then we have highlighted in the middle at line -- I guess
18 Line 9, code.parseInstruction. Do you see that?

19 **A.** So if you're going to simulate a course of computer
20 instructions, you're going to have to know what those computer
21 instructions are and what they say to do.

22 Parsing in computing refers to understanding the
23 parts of an instruction. So parseInstruction means figure out
24 the name of the instruction and the arguments to it so you can
25 figure out what it does. And here what we want to do with that

1 is find the data an associated with the instructions that
2 initialize the array. So we can write out the short
3 instruction that says initialize an array and here is the data
4 that goes in the array.

5 **Q.** So does the dx tool simulate execution of the Java
6 bytecode using parsing?

7 **A.** Yes.

8 **Q.** Now, then, we're look at Lines 948 to 950, and 887. What
9 is that -- what is that showing us?

10 **A.** So here is another method from another file. This is
11 related to parsing. And this method goes fluid for array
12 initialization and follows the sequence of instructions, one at
13 a time, to look for the sequences that are needed in order to
14 do this optimization and transformation.

15 **Q.** So what is the comment there that's highlighted?

16 **A.** So this is:

17 "Try to match the array initialization
18 itinerary idiom."

19 So "idiom" refers to the way the compiler typically
20 or always produces array initialization sequences.

21 **Q.** And then it goes on --

22 **A.** And says, well, when that occurs there will be this
23 pattern of instructions. Dup, push, push and store are Java
24 bytecode instruction names.

25 **Q.** In your opinion, Dr. Mitchell, are we simulating here?

1 **A.** Well, with c in the source code that actually shows how
2 this works, that, yes, we are.

3 **Q.** Okay. Let's go on to Lines 887 through -- and then 963
4 through 980. What is this source code telling us?

5 **A.** Well, this is part of the parseNewarray method. And here
6 you can -- we can see this code steps through the instructions
7 one at a time, looks at what they are, and eventually collects
8 the data that's contained in these instructions in order to
9 write the short array initialization command.

10 So this is simulation. It walks through the
11 instructions one at a time. It keeps track of the information
12 it needs in order to produce the succinct instruction, but it's
13 not full execution. If we wanted to execute it, that wouldn't
14 work here. We really just need to do something similar to
15 execution, simulate it, to get the effect we need, which is the
16 set of data values that are used to initialize the array.

17 **Q.** Back to my grocery store analogy and the grocery list.
18 What's happening to the grocery list here in the code?

19 **A.** This is kind of thinking through the process without
20 walking around the store.

21 **Q.** So then we come to some more lines of code. We're in
22 parseNewarray. So we have been seeing that at the top.

23 Now we're at 982 through 992 of TX 46.7. What is
24 that showing us?

25 **A.** This shows that the main part of this here is it's finding

1 out what values are used in initializing the array and the
2 bytecode.

3 And at the end it shows how an array is accumulated
4 within the dx tool that stores the values that go in the static
5 array that's being defined in the bytecode. And that's
6 important because writing this fast instruction requires the
7 list of data values that go into the array eventually.

8 **Q.** So did you do an experiment to confirm the way -- to
9 confirm dx operation in Android?

10 **A.** Yes. This looks a lot like the patent illustration and,
11 also, Dan Bornstein's example. But I just wanted to do it
12 myself just to make sure that I understood fully how this
13 works.

14 So here is a source code that defines a static array.
15 I called this Test. And it has 10 values, just the numbers one
16 through 10.

17 I compiled this, and the next slide shows what the
18 compiled bytecode looks like, and this is just like the patent
19 description. This is the output of the Java compiler on source
20 code that initializes a static array.

21 **Q.** And, again, why are you using the Java compiler for this
22 when we're looking at Android?

23 **A.** This is the way that Android applications are built, too.
24 Start with Java source code, compile it with the standard Java
25 compiler, and then apply an Android specific tool after that.

1 Q. So how many -- and how many entries did your source code
2 defining an array with elements 1 through 10, how many entries
3 were required by the Java bytecode output of the Java compiler?

4 A. Well, they use all the 10 array entries. The bytecode is
5 about -- you know, looks like 50 some-odd instructions in order
6 to do that.

7 Q. So what did you next do and learn?

8 A. Then I ran the dx tool. This is part of the Android SDK
9 or software development kit, runs dx on the output of the
10 compiler. And then there are various tools that you can use to
11 display them, the DEX file.

12 And here, looking at the DEX file, this just looks
13 like Dan Bornstein's slide in effect. There is a
14 fill-array-data bytecode. I don't think you could be much
15 clearer with a short bytecode name than that. This fills array
16 with data. And then the data is needed.

17 And the way it's represented here is in the --
18 effectively in the instruction stream beyond the return, there
19 is just the list of the indices and the values that go in those
20 places in the array.

21 Q. What did this confirm for you?

22 A. This shows that the dx tool takes the bytecode, applies a
23 process that from the source code we can see is a simulation
24 process, and uses that to produce a succinct instruction
25 filling an array with data and the exact data needed in order

1 to fill the array with data correctly.

2 **Q.** So let's sum up. The disagreement is about whether
3 Android simulates execution of the bytecodes. And what did
4 your review of the code demonstrate to you on this question?

5 **A.** I'm not sure what else to call this besides simulation.
6 This slide shows the claim language about simulating execution.
7 And then this is a summary of the source code that does the
8 simulation.

9 There is a class called Simulator, a method called
10 Simulate. And as expected, in order to simulate the execution
11 of code, you need to parse the code and step through it in some
12 form in order to see what it does and collect the data
13 arguments from the code in order to produce the shorter
14 instruction that does this more efficiently.

15 **Q.** So does the dx tool simulate execution of the bytecodes of
16 the clinit method against a memory without executing the
17 bytecodes to identify the static initialization of the array as
18 set forth in Claim 1 of the '520 patent?

19 **A.** Yes, it does. The source code simulates the execution,
20 sees what it -- sees what that execution would do against the
21 memory and -- but it doesn't execute it. We can see that it
22 just simulates what these instructions would do. And
23 correctly, as the examples show, it identifies static
24 initialization of array.

25 **Q.** What functions does the simulator use to accomplish that?

1 I'm on Slide 70 now.

2 **A.** So it uses parseInstruction and parseNewarray methods.

3 **Q.** Now, did you look at Claim 20?

4 **A.** Yes. This is similar and it's infringed in the same way.

5 **Q.** Claim 20 is a dependent claim. We heard a little bit
6 about that in the -- in the early part of this phase. Can you
7 explain how to understand this claim?

8 **A.** Yes. So it's a little bit harder to read and understand
9 because of that structure, but what Claim 20 says is that the
10 computer readable medium of Claim 18.

11 So I think we can just read Claim 20 as follows, as
12 if it's added to Claim 18. We just look to make sure that all
13 of the conditions and limitations of Claim 18 are met, and in
14 addition add the limitation of Claim 20.

15 **Q.** And what's the dispute on Claim 20?

16 **A.** The dispute has to do really with this, the word
17 "simulating."

18 "Simulating execution of the code or bytecode
19 to identify the static initialization of the
20 array."

21 **Q.** And what is your conclusion as to infringement of Claim
22 20?

23 **A.** Well, this source code says simulate. The way I
24 understand what it does, is that it performs a simulation. So
25 I see this as very clearly met on the basis of the way the code

1 works, the way the code is described, and my experiments to
2 look at how it operates.

3 **Q.** Professor Mitchell, do you have any indication that
4 third-party OEMs who are creating Android applications using
5 the dx tool have disabled the static array initialization
6 function in that tool?

7 **A.** No, I don't.

8 **Q.** Do you have any evidence that suggests they have not done
9 that?

10 **A.** I don't see why they would. It's a helpful thing. It's
11 built into the system.

12 It's also -- this is in the developer portion of the
13 platform. So even if for some reason a vendor wanted to modify
14 this, most developers for that platform would still be using
15 the SDK from Google.

16 **Q.** Is the performance -- how would you characterize the
17 performance benefit or memory savings arising from Android's
18 use of the '520 patent?

19 **A.** You basically get what you get. If you have a program
20 that uses large static arrays, then they are reduced and
21 then -- then the bytecode is reduced in size, exactly the way
22 we've shown by these examples. If the array was 10 times as
23 long, you would get 10 times the compression we see in these
24 examples.

25 For programs that don't have large static arrays

1 initialized in this way, there wouldn't be any effect. But
2 these arrays do occur and are used in applications and when
3 they occur, this is a very significant improvement for that
4 particular part of the code.

5 Q. Did you direct any performance analysis with respect to
6 the '520 patent?

7 A. Yes, yes. I worked with Noel Poore, who carried out some
8 experiments here.

9 Q. And what direction did you give him?

10 A. I asked him -- I don't remember all the details, but I
11 talked with him over some period of time and suggested that he
12 try build some programs to see what the difference was between
13 the bytecode file and the DEX file and how significant that
14 was.

15 Q. Did you review the results of that work?

16 A. Yes, I did.

17 Q. And what was your conclusion based on those results?

18 A. Well, basically, as I described, this is an important
19 improvement for programs where static initialization of arrays
20 of appropriate types of data are used.

21 Q. Now, did you see any indications in Google's documentation
22 that corroborated the performance analysis that you had
23 Mr. Poore do, the --

24 A. I don't recall --

25 Q. Mr. Bornstein's -- do you recall Mr. Bornstein's video on

1 this point?

2 **A.** Yes. I have forgotten the details of what he said, but he
3 selected this as something to summarize and highlight in his
4 presentation. And I believe he considered it valuable and
5 worth explaining.

6 **Q.** Now, back to the '104 patent.

7 Did you direct that performance analysis be done with
8 respect to the '104 patent?

9 **A.** Yes, I did.

10 **Q.** And what direction did you give there?

11 **A.** I wanted to look at the effect of both the dexopt and the
12 Resolve.c portions of the Dalvik Virtual Machine. And so I
13 suggested producing variants of the Dalvik runtime, which had
14 one or both of those things disabled, and then running to see
15 the performance of the Dalvik Virtual Machine with these
16 features removed.

17 **Q.** Now, when the engineers who did the performance analysis
18 were examined by Google's counsel, they were asked whether it
19 mattered that -- well, they were asked, first of all, whether
20 they actually did the performance analysis on Android phones;
21 do you recall that --

22 **A.** Yes.

23 **Q.** (Continuing) -- that back and forth?

24 Based on your knowledge and experience, are the
25 performance tests that the engineers did using platforms for

1 testing, are they in any way less probative, less valuable in
2 understanding the performance impact of these patents because
3 they were done on those platforms?

4 **A.** Well, I think the platforms that were used were the
5 standard kinds of test platforms that I understand are also
6 used by Google to evaluate their performance of the system as
7 they build and develop it.

8 Benchmarking really refers to running something in a
9 lab or separate environment to see how well it works with
10 particular input that a community of developers believe gives
11 meaningful or representative performance evaluation.

12 **Q.** Now --

13 **MR. VAN NEST:** Objection, your Honor. I move to
14 strike as to what Google does in the area of benchmarking.
15 That was not within the scope of the question.

16 **THE COURT:** All right. That part will go out.
17 Unless there is a fresh question, that part of the answer will
18 be stricken.

19 It's 30 seconds to 1:00 o'clock.

20 **BY MR. JACOBS:**

21 **Q.** Is there a relationship between performance and the
22 performance that was measured, for the '104 patent in
23 particular, in battery life?

24 **A.** Yeah. Every instruction requires some movement of
25 electrons inside the computer. The reason why executing a --

1 running a processor -- well, if you run fewer instructions or
2 have the processor active for less time, you use less battery.

3 **Q.** And is that important to handset makers?

4 **A.** I think we all have a probable every now and then of
5 forgetting to charge our phone or leaving it running or doing a
6 lot of things during the course of the day. The longer the
7 battery runs, the more useful a portable phone is.

8 **Q.** Based on the battery life relationship to the '104, what
9 does that suggest to you about whether handset makers would
10 have modified the code to take out the '104 functionality?

11 **A.** That's another reason why this is useful and valuable to
12 handset manufacturers to have the system work as well as
13 possible, as quickly as possible, and to drain the battery as
14 slow as possible.

15 So I don't see why a vendor, a phone manufacturer,
16 would want a less desirable system on their phone.

17 **THE COURT:** Is this a good breaking point?

18 **MR. JACOBS:** Yes, it is, your Honor.

19 **THE COURT:** We'll break here for the day. Please
20 remember the admonition. We will see you here at 7:45 in the
21 morning.

22 **THE CLERK:** All rise.

23 (Jury exits the courtroom at 12:59 p.m.)

24 **THE COURT:** Please be seated.

25 Professor Mitchell, we'll need you to be back

1 tomorrow normal time, 7:30. You have a good day. We'll see
2 you here then.

3 **THE WITNESS:** Thank you.

4 (Witness steps down.)

5 **THE COURT:** On the deposition of Mr. Rubin, what was
6 the break down on that.

7 **MR. JACOBS:** One minute to Google, your Honor. The
8 rest to us.

9 **THE COURT:** Okay. Well, both sides still have ample
10 time on this phase, but I'll give you the -- I'll give you the
11 break down later on.

12 We have a meeting at 1:45 to go over your Rule 50s,
13 right?

14 **MR. JACOBS:** Yes, your Honor.

15 **MR. VAN NEST:** Correct.

16 **THE COURT:** Anything you want to bring up with me
17 right now?

18 **MR. JACOBS:** I don't think so.

19 **MR. VAN NEST:** No, your Honor.

20 **THE COURT:** I have something I would like to bring
21 up, get your help on.

22 As I told you, I'm working on research and thinking
23 about your other issues concerning copyrightability. So I'm
24 going to hand down to both sides a draft of -- just a small
25 part of the -- I don't even have a full order prepared. I

1 don't even know the answer yet.

2 (Whereupon document was tendered
3 to counsel.)

4 **THE COURT:** So don't try to read anything into this.
5 You cannot blame my law clerk. This is all my fault. This is
6 my stab at reading your materials and trying to come up with a
7 description of the Java system that is short and sweet, but at
8 the same time builds into it the most important features that
9 will be important on appeal.

10 So what I'm going to ask you to do is two things. If
11 there is something that's actually wrong, I want to know that.
12 So you can correct anything that is wrong.

13 You can also add three sentences. You cannot add
14 four. In other words, I don't want you to use this as a way to
15 lard in your arguments. I'm not going to do that. That won't
16 be useful. But I am willing to entertain three or four more
17 sentences that you may think are important and that are a
18 neutral statement about how this thing works.

19 I'm intending this to be an honest non-argumentative
20 straightforward statement about how the Java system works for
21 which we have, you know, many thick books, and this is going to
22 be three short pages.

23 So you can see how difficult that challenge is, but
24 otherwise we would be sending up to the Court of Appeals a
25 monster that would do them no good.

1 So please, two things. If it's -- Mr. Baber, I hope
2 you're listening. If it's wrong, you can fix it. If it's
3 right but you don't like the way it's worded, too bad.

4 **MR. BABER:** Understood, your Honor.

5 **THE COURT:** And then you can add three sentences that
6 are non-argumentative and are, you know, straightforward
7 statements about how the system works. And then you can give
8 this to me at 1:45 when we come back.

9 **MR. VAN NEST:** I thought I heard you say three or
10 four sentences, your Honor.

11 **THE COURT:** Three.

12 **MR. BABER:** Three not four.

13 **THE COURT:** I will give you four.

14 **MR. VAN NEST:** Thank you. Thank you.

15 **THE COURT:** Ten words each.

16 (Laughter.)

17 **MR. VAN NEST:** Now you want a grade on this?

18 **THE COURT:** Ten words each.

19 No, I'm looking for something neutral,
20 straightforward.

21 Now, there is one thing that I could not quite figure
22 out in the difference between the word "expression" and
23 "statement." I would -- and one of your four can be to help
24 explain the word "expression." I don't mean "expression"
25 versus "idea" under the Copyright Act. I mean the word

1 "expression" as it's used in the computer language itself.

2 That would be a useful thing to add in here?

3 And I want you also to know that the -- the books are
4 not consistent with the way some of you use the word
5 "declaration." The book that's in evidence and which was
6 written by the pros seems to use the word "declaration"
7 slightly differently than was -- and maybe even the word
8 "signature."

9 I would like for you to help me understand what the
10 correct answer is so that we can send it up to the Court of
11 Appeals with a -- you know, a deck of cards that has 52 cards.
12 You each know what's in there and you make your arguments based
13 on that and have a common body of description here that works.

14 So there we are. I will see you back here at 1:45.
15 Thank you.

16 (Whereupon there was a recess in the proceedings
17 from 1:04 p.m. until 1:45 p.m.)

18 **THE COURT:** Please be seated. Let's go back to work.

19 **MR. JACOBS:** We have a joint request.

20 **THE COURT:** Okay. Make sure we're hooked up. Are we
21 ready to go, Katherine?

22 We're ready. Okay. Joint request.

23 **MR. JACOBS:** We're both working hard on the Java
24 description, but would request to 5 o'clock to get it back to
25 you.

1 **THE COURT:** That's fine. No problem.

2 I have a related request, but you tell me if this is
3 too hard to do.

4 I think it would be useful to have a chart that had
5 the 37 packages down column 1. Next column would be number of
6 classes in package Java.

7 Next would be number of methods. And here I need
8 your help. Methods, interfaces and fields, or methods alone
9 would be, perhaps, enough. But methods and interfaces Java.
10 And then the same two columns but for Android.

11 So it would be a 5-column chart, 37 rows with titles.
12 And it would indicate the number of classes, the number of
13 methods broken out by each of the 37.

14 Now, you ought to be able to reconstruct that just
15 from the code itself. And the code itself is in evidence,
16 right?

17 **MR. JACOBS:** Yes, Your Honor.

18 **THE COURT:** So would that be a doable project, or is
19 that just too much for two gigantic companies with seven or
20 eight lawyers at each table?

21 (Laughter)

22 **MR. JACOBS:** Very doable, Your Honor.

23 **MR. VAN NEST:** Is that a rhetorical question, Your
24 Honor?

25 **MR. JACOBS:** It's very doable, Your Honor.

1 **MR. BABER:** It's not only doable, Your Honor, I think
2 it's already been done. The expert reports -- I believe,
3 Professor Astrachan's report, he has a chart very much like
4 that.

5 **THE COURT:** Well, I would like to see it, but I -- if
6 you two are going to start arguing over the -- I'd like for you
7 to iron out your -- this is a matter of counting up the items.

8 So, anyway, if you could get me something like that
9 by tomorrow, that would be good.

10 **MR. JACOBS:** Thank you, Your Honor. We will do that.

11 **MR. KWUN:** Your Honor, there is one technical point.

12 There's two different ways you could count the number of
13 methods that are in a class. One is, you could count the
14 number that are actually declared in that class. But as Your
15 Honor knows, you can also inherit methods from a super class.

16 So to the outside world, it doesn't matter whether
17 those methods were declared within that class or inherited from
18 a super class. So you could either count the number of methods
19 that are sort of available to that class, or you can count the
20 numbers that are expressly declared.

21 **THE COURT:** I want the expressly declared.

22 **MR. KWUN:** Thank you, Your Honor.

23 **THE COURT:** And you can put in an asterisk on that
24 point. But the -- kind of like counting up the number of lines
25 that declare something.

1 Okay. We're here for Rule 50. And we have motions
2 on both sides. Let's start with the motion for JMOL by Oracle
3 on fair use.

4 **MR. JACOBS:** I'd like to start, Your Honor, by giving
5 you a list of cases from 1992 to the present, that have found
6 no fair use as a matter of law.

7 And we begin with *Triad*. These are Ninth Circuit
8 cases. 64 F.3d 1330. That was a case involving an independent
9 service operator copying code into RAM.

10 And the Court decided that that was
11 nontransformative, and on factor four noted that:

12 "If independent service operators like
13 Southeastern freely used *Triad's* copyrighted
14 software on a widespread basis to compete
15 with *Triad*, this would likely cause a
16 significant adverse impact on *Triad's*
17 licensing and service revenues, and lower
18 returns on its copyrighted software
19 investment."

20 Then there's the *Wall Data* case, in which the Court
21 held that while software was not purely creative, it is
22 protected under the Copyright Act. The plaintiff presented
23 undisputed evidence that its software products were developed
24 over several years, required a multi-million-dollar investment.

25 And, on factor four:

1 "Whenever a user puts copyrighted software to
2 uses beyond the uses it bargained for, it
3 affects the legitimate market for the
4 product. And widespread unlicensed
5 copying" -- which would be the effect of a
6 fair use ruling in the defendant's favor in
7 *Wall Data* -- "could affect the market for the
8 plaintiff's software."

9 So that's 447 F.3d 769.

10 So, again, these are cases deciding against fair use
11 as a matter of law.

12 Then there's the *Worldwide Church of God* case, which
13 is interesting in that it falls into the category that I think
14 we could box this case into. It's kind of an adjacent markets
15 case. "A" has a copyrighted work. It is distributing it,
16 licensing it, marketing it in a certain realm. "B" comes along
17 and takes the heart of the copyrighted work and decides to
18 market it in a new way, in a new form, to a new audience. It
19 doesn't fundamentally change it.

20 And *Worldwide Church of God* was a case about books
21 that had been appropriated for use in a different church. So
22 an entirely different audience. And no fair use was found
23 there.

24 The Court noted there that:

25 "If there are no genuine issues of material

1 fact, or if even after resolving all issues
2 in favor of the opposing party, if a
3 reasonable trier of fact can reach only one
4 conclusion, a court may conclude as a matter
5 of law whether the challenged use qualifies
6 as a fair use."

7 And citing cases that did that as a matter of law.

8 L.A. News is another interesting case for the kind of
9 adjacent markets theory that I was describing. 149 F.3d 987.

10 That's a case where the plaintiff owned the works.
11 They produced videotapes and licensed them to a broadcasting
12 company. When the broadcasting company aired the works, it
13 simultaneously transmitted them to a television news agency with
14 which the broadcasting company had an agreement. And defendant
15 copied the works and transmitted them to paying subscribers.

16 Summary judgment in favor of plaintiff, finding no
17 fair use, was commercial, not nonprofit, not very
18 transformative. They took the heart of the work.

19 "Allowing such use would result in a
20 substantially adverse impact on the potential
21 market for the original work."

22 So the potential market.

23 I think there's an important point here. We do not
24 have to prove lost profits to prove an adverse impact on the
25 market for the Java software that is threatened by Android.

1 This is not that kind of causal nexus.

2 Courts frequently make matter of law predictions
3 about the impact of a defendant's activities on the
4 plaintiff's -- on the market for the plaintiff's work.

5 The *Leadsinger* case. I mentioned this earlier in our
6 discussions. This is 512 F.3d 522. A 2008 Ninth Circuit case.

7 This is at the motion to dismiss -- this is affirming
8 a motion to dismiss. The defendant sought declaratory judgment
9 that it could copy song lyrics into karaoke. That's kind of
10 interesting because the song lyrics are only a component of a
11 song, and they are arguably being placed into a new -- again,
12 an adjacent market.

13 But this case didn't even get beyond the pleading
14 stage. There was no claim of transformative use. Karaoke does
15 not add to or alter the copyrighted lyrics and is not
16 transformative.

17 And then that brings us to the *Abend* case, which in
18 some ways is the most interesting of all because *Abend* owned
19 the copyright on the original story for *Rear Window*, the
20 Hitchcock movie. And *Rear Window* was more or less out of
21 distribution. And MCA did a re-release of the film in
22 theaters, on TV, and on video cassette.

23 And the District Court granted summary judgment for
24 the defendants based on fair use. And the Court of Appeals
25 reverse.

1 "Commercial use of a fictional story that
2 adversely affects the story owner's
3 adaptations rights is a classic example of
4 unfair use."

5 Now, I say that's interesting because you can imagine
6 how the defendant there would argue analogously to Google's
7 argument.

8 All we did was take -- remember this, the story right
9 that's been infringed. So it's the structure, sequence and
10 organization of the *Rear Window* movie. It's the plot outline,
11 not the movie itself. The owners of the movie rights were not
12 the plaintiff. It was the story owner.

13 And the defendant there says, this is great for the
14 story. Look how many more people are going to see the story
15 underlying *Rear Window* if this movie is out in these new media,
16 in video cassettes or just re-released into theaters.

17 The court did not give that argument much weight.

18 Applied here, these cases allow for only one outcome
19 on Google's fair use defense.

20 Going through the factors: Google's use is purely
21 commercial. There is no nonprofit purpose. This is no
22 educational purpose.

23 Of course, they haven't taken the APIs and subjected
24 them to programmers' criticism. They have not done anything
25 other than port the APIs and the structure, sequence and

1 organization of the code into Android, and deployed it into a
2 market close to -- arguably already occupied by Java, in that
3 Java is on smart phones -- so close to and a market in which
4 Java could reasonably expect to be deployed.

5 How do we know that? We know that because there was
6 licensing negotiation between Google and Sun, in which Google
7 sought to take the Java Application Programming Interfaces, the
8 SSO, et cetera, and deploy it into this adjacent market into a
9 slightly -- and using a different licensing model.

10 So they sought a license. There's facts going both
11 ways on what license they sought for what when. But the fact
12 of the matter is, the Java technology was so relevant as is to
13 Google's proposed market, that the parties spent a considerable
14 period of time trying to negotiate a license for that use.

15 And Google went ahead and used the valuable APIs
16 anyway, in their commercial product, never taking that license.

17 So we know exactly what market has been interfered
18 with here. There's no hypothetical. There was a licensing
19 discussion. And then, of course, there's lots of testimony
20 undisputed about Sun, now Oracle's, licensing model. The
21 specification license in particular being the license for
22 independent implementations. The most applicable license for
23 what Google did or wished to do.

24 If Google's use is fair use, that licensing market is
25 destroyed. There is no copyright right to back up that kind of

1 openness, that kind of freedom to develop independent
2 implementations.

3 There is no way, to use Mr. Schwartz's language, to
4 force compliance with this form of openness through the
5 assertion of copyright.

6 And then that brings us to the other kind of license
7 that was most -- that is most seriously disrupted here, and
8 that's the GPL. Because, of course, there's undisputed
9 testimony that Java is available pursuant to an open source
10 license.

11 It's an open source license that didn't suit Google's
12 commercial needs. And so they took the 37 packages and
13 deployed them in their own -- to their own end, using their own
14 license.

15 Once again, if that is fair use, there is no
16 copyright right to enforce the GPL -- to enforce GPL license
17 compliance, at least against someone who chooses to take only
18 the structure, sequence and organization. Which, as we've all
19 heard, is the heart of the matter here. It's the most valuable
20 part of the copyrighted work as a whole. It is the 37 packages
21 that Google wanted because they were the most popular.

22 So this is another reason why this cannot be fair
23 use. Popularity does not allow for infringement. Investment
24 does not allow for infringement.

25 It can't be the case that the more popular your work

1 is, the more the defendant can take it and deploy it to its own
2 commercial benefit. That just gets copyright law and the
3 incentives in the Constitution backwards.

4 There is case after case, and it is valid, strong
5 authority today that a commercial use is presumptively unfair.
6 The cases that have gone the other way on that question have
7 been categorical -- have been cases in categories.

8 Parody cases, in which the parody itself is
9 commercial. But we like parodies, and we think they are
10 valuable, and they don't really destroy the market for the
11 original work. Because if you want to see the original work,
12 you've got to see the original work not just the parody.

13 There are the reverse engineering cases that we've
14 discussed at length in our brief. Again, a kind of a category
15 in the law where the ultimate product is concededly
16 non-infringing. And the only question is whether the
17 intermediate copying is excused on fair use grounds. Again, a
18 very narrow category.

19 But *Harper & Row*, *Passport Video*, the *Leadsinger* case
20 as recently as a couple of years ago, all reinforce that a
21 commercial use -- in the general case, a commercial use is
22 presumptively -- gives rise to a presumption of harm to the
23 market and, therefore, the other factors are going to have to
24 tilt very heavily in the defendant's favor if we're ever going
25 to find in favor of the defendant on fair use.

1 Looking more narrowly at the doctrine around
2 transformative, Google's use was not. The transformative cases
3 are trying to address a case in which the expression is recast
4 or recapitulated in a form that changes its underlying message.

5 And probably Google's best case is a Google case, in
6 which the plaintiff owned photographic rights. Google took
7 thumbnails of them as part of the search engine. And so Google
8 wins that a search engine use of thumbnails of the original
9 photos is fair use. Why? Well, a search engine is an entirely
10 different purpose, an entirely different use.

11 Now, frankly, I think that's a pretty close case.
12 The idea that you could reproduce a photograph in thumbnails
13 and display it to the world without getting a copyright, that's
14 a close case under fair use. But, it did go Google's way.

15 But just compare the facts there with the facts here.
16 A search engine as compared to a photographer. Android as
17 compared to Java. A software platform as against a software
18 platform. Couldn't be software platform to software platform
19 can't be much closer. And photograph to search engine
20 considerably more remote.

21 So Google did nothing to change the message of the
22 APIs. The APIs were a popular, attractive, heavily-invested-in
23 way to reach programmers and make the Java platform attractive
24 as a programming environment. And looking in the other
25 direction, a set of design materials for class library

1 creators, creating independent implementations under the
2 specification license.

3 What did Google do? Took the 37 packages. They
4 created very similar documentation. They put it out there for
5 Java programmers to use in the same way that they use the 37
6 packages in Java, and created class libraries. Independent
7 implementations, they claim. We saw that wasn't true, but it
8 doesn't really matter for present purposes.

9 They created supposed independent implementations in
10 core libraries, just like the licensed or Sun developers do
11 with that information.

12 So there is no recasting. There is no reforming.
13 There is no new message. It's the very same message to the
14 very same purposes -- purpose, for purposes of fair use
15 analysis.

16 Creative versus functional. One can debate this.
17 Obviously, software is not a symphony. Software is not a poem.
18 But what was interesting about this trial was the undisputed
19 evidence from both sides and both sides' experts about how
20 creative the authoring process of APIs is.

21 This is not the creation of a functional work within
22 the meaning of copyright law. Not when we heard from witness
23 after witness how much flexibility there is in creating APIs,
24 and how much creative labor went into the process of creating
25 these 37 Java APIs.

1 Now, let me -- let me just do a parenthetical here,
2 because I think it's important.

3 From what you have signaled to us so far, Your Honor,
4 the decision you're writing is going to be heavily grounded in
5 the facts. The facts of Java. The facts of the Java
6 Application Programming Interfaces and these 37 packages.

7 Not all interfaces are created equal for purposes of
8 copyright or for purposes of fair use analysis.

9 These core library package APIs are very closely
10 drawn to the underlying code itself. We saw that in the method
11 declarations showing up in the API documentation and in the
12 code.

13 This is not merely a set of -- of numerical values
14 that represent an interface to a PlayStation box or a Nintendo
15 box. Thousands of pages of writing represent the 37 packages
16 here.

17 So the word "interface" shouldn't be overused in
18 understanding these fair use cases. We're talking about these
19 37 packages and the creativity and authorship that went into
20 them, not whether every computer program is creative or
21 functional. Not whether every interface is creative or
22 functional. These programming interfaces.

23 Undisputed evidence about the creativity that went
24 into the authoring process.

25 Third factor, amount and substantiality of the

1 portion used.

2 *Harper & Row*. Great case. 300 words out of 200,000,
3 and just a week before the book is coming out. So all sorts of
4 arguments: We made a bigger market for the book. We have more
5 readers for the book because we're a magazine and we're the
6 nation and we're publishing an extract from *Harper & Row*.

7 But, no, you can't take 300 words out of 200,000 from
8 an unpublished Presidential Memoir. The portions actually
9 quoted were selected, according to the Supreme Court, as among
10 the most powerful passages in those chapters.

11 And that's what Google did here. They took the 37
12 packages. They took -- one of their witnesses said, We took
13 the good stuff.

14 They took the 37 packages that they thought the
15 programmers programming for Android would most want to see, and
16 left the rest behind.

17 And then, finally, that brings me back to the harm to
18 the market. And their only evidence, if you call it that, on
19 Google's side is that Java ME is doing okay under Oracle.

20 Well, a couple of things about that. First of all,
21 undisputedly, Java ME was not the target for Android. Java SE
22 was.

23 Secondly, the undisputed evidence is that Java ME is
24 not doing well in Android's markets. Java ME is doing well in
25 places where Android has not yet penetrated.

1 And, by contrast, Java ME was licensed, for example,
2 to Amazon for its Kindle, and is no longer licensed to Amazon
3 for the Kindle Fire because Android has taken its place.

4 So even in the most kind of granular causation-based
5 analysis, much more rigorous than the cases require for fair
6 use, we have evidence of direct market supplantation.

7 But I think -- but, the evidence was undisputed,
8 again, that the Java model is a comprehensive model of
9 licensing of both underlying code and of the specifications;
10 that that model has been threatened by Android because Google,
11 notwithstanding the previous negotiations, took what it wanted
12 for itself without a license.

13 The whole approach that Sun, now Oracle, takes to
14 fragmentation is threatened by Android because Android took
15 subset and superset it.

16 And while there was a lot of evidence going both ways
17 on fragmentation -- on whether the word "fragmentation" can
18 reasonably be applied to say the differences between ME and SE,
19 there was undisputed evidence that within platforms Sun, now
20 Oracle, took aggressive measures to try as best they could to
21 control this kind of open source model in which there are
22 independent implementations.

23 And, it's undisputed that Oracle, upon acquiring Sun,
24 invested substantially more resources in Java. There was
25 testimony from Mr. Reinhold about a near doubling of the number

1 of engineers working on Java. So the commitment is evident in
2 the assignment of resources to the project.

3 And what Google is doing is looking backwards and
4 saying, well, Sun was not a strong company. Sun had let
5 fragmentation develop.

6 That's not the right analysis. They don't get to
7 take -- this is not a case where -- this is not an area of law
8 where the defendant gets to take advantage of the plaintiff's
9 weaknesses at a particular moment in time and then say, see, we
10 get away with it.

11 So on all four factors, Google loses. Loses
12 strongly. And, again, on undisputed evidence.

13 But the underlying purpose of copyright law and the
14 underlying purpose of fair use also needs to be looked at here.
15 One of the basic reasons for fair use analysis is to deal with
16 a situation in which, by its nature, the plaintiff would be
17 unwilling to grant a license. That's the parody case. That's
18 why we have a commercial use for -- commercial use is okay even
19 if it's a parody because most authors don't like their works to
20 be parodied.

21 But here there is a license. Google may not like it
22 for business reasons. May have had a different business model
23 in mind. That was Mr. Rubin's testimony. The negotiations
24 broke down because Sun would not adopt Google's business model.
25 But business model differences do not give rise to fair use

1 defense.

2 Thank you, Your Honor.

3 **THE COURT:** All right. Google's turn.

4 **MR. VAN NEST:** Good afternoon, Your Honor.

5 I think the cases make very clear that commercial use
6 is not a disabling factor. It's simply one factor.

7 All of the leading fair use cases recently in our
8 Circuit have dealt with situations where the defendant's use
9 was a commercial use.

10 If you look at *Sony vs. Connectix*, the video
11 equipment manufacturer engineered Sony's APIs and created a
12 competing game platform to play video games.

13 *Sega vs. Accolade*, video game maker copied Sega's
14 APIs and made video games that were compatible with the Sega
15 system.

16 *Campbell*, recent Supreme Court case was a rap
17 group --

18 **THE COURT:** Sorry, did the *Sega* case involve APIs? I
19 didn't remember that part.

20 **MR. VAN NEST:** They had to reverse engineer the
21 interfaces so they could make games that were compatible. They
22 are not the same kind of APIs we're talking about here.

23 But what they ended up making was video games that
24 competed with the video games Sega sold for the Sega platform.

25 **THE COURT:** Well, of course, I understood that part.

1 But you said they copied the APIs. I don't even remember that
2 term being in the decision.

3 **MR. VAN NEST:** They --

4 **THE COURT:** Are you sure that term is in the
5 decision?

6 **MR. VAN NEST:** I'm not sure, Your Honor. But what
7 they did was reverse engineer the Gateway, if you will, to the
8 Sega system so they could make games that were compatible with
9 it.

10 My point is simply that all these cases -- including
11 *Campbell* in the Supreme Court, which says there is no
12 presumption against fair use just because there's a commercial
13 use. *Campbell* was a rap group selling its music for profit.

14 So all of these cases confirm that, just as Your
15 Honor's instruction says, the commercial use is one factor.
16 And fair use determination involves weighing all the factors,
17 and giving them the weight that the jury deems appropriate.

18 Similarly, on the other big point for Oracle, the
19 mere fact that the defendant's use has some impact on the
20 plaintiff's copyrighted work also doesn't disable anything.

21 Obviously, in Sony, the whole point was to make a
22 competing platform. Connectix did that so that people could
23 play games on a computer that they were currently only playing
24 on Sony's PlayStation.

25 So the court there said, very clearly, that, yes, we

1 understand there's going to be an impact on Sony. But we
2 recognize that this is a brand-new platform that's been
3 created; that the whole point of the copyright law to protect
4 expression, not ideas. If people are creating a new platform
5 where others can express themselves and compete, that's --
6 that's consistent with fair use, consistent with the purpose of
7 the Copyright Act.

8 And, obviously, in Sony there was no question there
9 would be an impact on the plaintiff, and their fair use was
10 determined to be appropriate.

11 Same with Sega. In Sega there was no question that
12 both companies were going to be in the business of selling
13 games, and there would be competition and impact.

14 So there again the Court said, because it's a
15 functional nature of -- it's a computer code here, we're going
16 to -- we're going to allow fair use.

17 Now, I would say the cases they are relying on, most
18 of the ones Mr. Jacobs referred to are not in their brief, but
19 they are all cases about things like books, where the book is
20 wholesale copied, or poems, or songs, or movies, or plays.

21 This is not such a thing. This is a situation where
22 we're talking about computer software, which is purely
23 functional.

24 When you get down to looking at the four factors,
25 there is an overwhelming amount of evidence of fair use,

1 consistent with what we told the jury in the opening and the
2 closing.

3 And I'm not going to claim that there's anything
4 undisputed in this case. There was plenty of evidence adduced
5 by both parties.

6 But with respect to the transformative nature of
7 Android, there were two kind of principal points. One was, it
8 was an open platform. The platform was not licensed or sold.
9 It was open for anyone to use.

10 And Your Honor heard testimony from Rubin, from
11 Schwartz, from Ellison, that they all attempted to take
12 advantage of the Android platform.

13 It was out there for everybody. It has fostered
14 increased expression in the form of more application
15 developers, increased competition among the handset makers, the
16 carriers, and the app developers. And both Sun and Oracle had
17 an opportunity to compete.

18 This is exactly what the Ninth Circuit held was
19 appropriate in *Sony vs. Connectix*. If you are opening up a new
20 platform, that is consistent with fair use.

21 It was also, point two, a brand-new product.

22 Your Honor is aware that the testimony was the 37
23 APIs were incorporated into a full stack. The folks at Sun had
24 not been able to create such a full stack, although they are
25 the experts on Java.

1 You heard testimony that the 37 APIs interact with
2 other layers in the stack, particularly the application
3 framework. That all the source code, all the implementing code
4 in these 37 packages was brand-new and totally different, other
5 than the nine lines we may talk about later this afternoon.

6 The platform supports all sorts of new functionality
7 and, therefore, there was more than enough evidence to find a
8 transformative nature. And the case law doesn't say that you
9 have to be using the functional features like this in a
10 completely new and different way.

11 Your instruction was quite right; what has been
12 added, what has been changed, and that's exactly consistent
13 with what we proved with respect to Android.

14 With respect to the nature of the copyrighted work,
15 no question it's functional. No question that to some degree
16 the words and names are necessary to run existing code. If you
17 want to run existing Java language code, you have to use the
18 same fully qualified names. You heard that from a number of
19 witnesses.

20 These APIs are expected to be available for
21 programmers in Java. Dr. Bloch testified at length about that.
22 Dr. Astrachan, too.

23 There are some of them that are absolutely required
24 just to run the language. There was a lot of testimony about
25 that and admissions by Dr. Mitchell and Dr. Reinhold that some

1 of these APIs are necessary just to use the language itself.

2 And it's clear in the Ninth Circuit that computer
3 systems and APIs are deemed functional and accorded less
4 protection as a result. There really is no doubt. I think the
5 testimony was that there was a lot of creative effort that went
6 into these. That's all fine and well, but the factor with
7 respect to the nature of the copyrighted work is addressed to
8 what's the nature of the completed work? And the nature of it
9 here is it's functional.

10 With respect to the amount of copyrighted material
11 used, again 7,000 lines out of 2.8 million lines, that was the
12 conclusion of both Dr. Astrachan, Dr. Mitchell.

13 Dr. Reinhold conceded that the SSO, which was all
14 that was at issue at the end of the day, was 7,000 to 10,000
15 lines of code out of 2.8 million.

16 The 37 API packages, what was used were the names and
17 the declarations. The source code was completely new,
18 completely different; 15 million lines of code, according to
19 Dr. Astrachan and Mr. Bornstein.

20 With respect to harm, I think there was -- there was
21 a plentiful amount of evidence here on both sides. Certainly,
22 there was conceded evidence that Java language is still number
23 one in the world. Dr. Mitchell testified to that. Java
24 profits at Oracle are up 10 percent year over year. Mr. Risvi
25 testified to that. According to Oracle, they are having

1 continued success with Java products and products like Rim and
2 Nokia. Mr. Screven and Dr. Reinhold testified to that.

3 And there is -- there may be evidence of a threat of
4 fragmentation, but no evidence of real fragmentation. Nobody
5 is confused that Android is a Java Platform or part of a Java
6 family.

7 And we heard a lot of evidence from Dr. Reinhold that
8 fragmentation now means limited to one platform, not across all
9 platforms. Well, Android is not the same platform at J2ME or
10 J2SE. It's a different platform.

11 **THE COURT:** Does the Android literature in any way
12 say that these programs previously written in Java will run on
13 Android, or is that just left to the developer to figure out on
14 their own?

15 **MR. VAN NEST:** I don't know, your Honor, if the
16 literature says that. I'd have to check. Given that we don't
17 have the Java brand, I doubt it, but I'm not sure. I don't
18 know the answer to that.

19 But I think developers are aware, as Dr. Bloch
20 testified and Mr. Bornstein, that the -- since you're writing
21 in the Java language and since people do know that Android is
22 written in the Java programming language, that that
23 functionality would certainly be available.

24 I think the other thing on potential market harm, I
25 do think here the testimony from Schwartz was also important.

1 He made a decision to welcome Android. They made a decision to
2 support platforms that support Android. They felt it would
3 have been a disaster if Android had used Microsoft language or
4 some other language. And they were persuaded, after debating
5 it internally, that Android could be a good thing for Java,
6 would be a good thing for Java. And, hence, that testimony
7 goes to a lot more than the equitable defenses. It also goes
8 to the fact that the people running Sun at the time thought
9 that Java -- that Android could place a set of rockets onto
10 Java. And they said that publicly.

11 They've used their own products on top of Android.
12 As your Honor heard, the JavaFx product was something they
13 featured at the JavaOne developer conference in 2008.

14 **THE COURT:** Repeat that again.

15 **MR. VAN NEST:** The folks at Sun not only endorsed
16 Android, but they built their own product to run on the Android
17 Platform. That's the product that was developed in the video
18 that we saw of the JavaOne conference in 2008. That's folks
19 from Sun on stage showing off the use of a Javafx product on
20 top of the Android Platform.

21 And there was discussion about that between Mr.
22 Schmidt and Mr. Schwartz. There was discussion about that
23 between Mr. Gupta and Mr. Rubin. There was a demonstration at
24 JavaOne in 2008.

25 And, again, it's all consistent with what Schwartz

1 said, which was: We knew that there were choices. We knew
2 that Android could use other languages. We felt it would have
3 been far, far worse for Android to go off and use some other
4 language than Java where we, Sun, wouldn't have any opportunity
5 to participate. And so that's the way it went.

6 The final point, your Honor --

7 **THE COURT:** Let me ask a question on this, on that
8 point.

9 Prior to the time that Oracle acquired Sun, what
10 emails or internal materials are there in the Sun records? Are
11 there some? It seems like there was something.

12 **MR. VAN NEST:** Yes.

13 **THE COURT:** No, but where they were saying it is
14 harmful, that Android is harmful to Java. Is there something
15 like that or am I thinking of the wrong thing?

16 **MR. VAN NEST:** I'm not --

17 **THE COURT:** Publicly you've got the rockets blog,
18 okay. That works in your favor. But were there some internal
19 documents that contradicted that?

20 **MR. VAN NEST:** I'm not aware of any documents that
21 were sent to Google that contradict that.

22 **THE COURT:** No, no.

23 **MR. VAN NEST:** But I'm not sure what --

24 **THE COURT:** I'm talking about internal.

25 **MR. VAN NEST:** There may have been some. I'm not

1 aware, but I would say this.

2 The rockets is just the beginning of it. Remember,
3 that at the same time as the blog, there was a personal email
4 from Schwartz to Schmidt saying, "What can we do to support
5 your announcement? We want to support your announcement."
6 That happened around the time in November.

7 Then they have a meeting in roughly March or April of
8 '08. Schmidt and Schwartz meet and there are emails around
9 that. Schwartz asked Schmidt: Can we build a product on top
10 of Android? We would like to do that. Can you show me your
11 licensing? What's the open source license? The Apache
12 license. So there is an email exchange between the two of them
13 where Schmidt sends Schwartz an email around the time of their
14 meeting reflecting they can do it.

15 Then there is another meeting between Rubin and Gupta
16 where Gupta comes to congratulate Rubin on the launch of
17 Android and says: We would like to explore building our own
18 JavaFx product on top of Android. Can we do that?

19 Then there is the JavaOne conference in '08 where
20 they demonstrate on stage, in the video we all saw, that they
21 have a JavaFX product running on Android.

22 All of that is happening. And the point of it is
23 simply that the folks running Sun at the time were trying to
24 use Android, the platform. They saw benefit in it for them.
25 They saw benefit in making positive statements about Android

1 and its usage of Java and they were attempting to participate
2 in the platform.

3 The final point I want to make, your Honor, is that,
4 I mean, the fair use law is pretty clear that -- and your
5 instructions reflect it absolutely; that the jury's job is to
6 look at all the factors. No one factor is determinative. They
7 are to be weighed together. The jury is to give them the
8 weight that they feel they deserve. And no one factor alone is
9 determinative.

10 And based on that, given the amount of evidence that
11 exists on each of the four factors, JMOL is simply not
12 appropriate.

13 Mr. Baber wants to add a comment or two.

14 (Brief pause.)

15 **MR. VAN NEST:** He's pointing out that in the Sega
16 case there is a reference to interface specifications at
17 Page 1515.

18 The defendant decompiled object code to get interface
19 specifications, then used the specifications and included
20 functional descriptions of interface requirements in their own
21 manual, but they wrote their own procedures to be compatible.

22 So it's a similar situation where in Android the
23 interfaces may be the same, but the source code implementing
24 those is different, original written by Android, written by
25 Google developers.

1 So that's an additional point on the *Sega* case.

2 **THE COURT:** All right. Very brief rebuttal.

3 **MR. JACOBS:** It's important to emphasize who bears
4 the burden of proof here. It's an affirmative defense. Google
5 bears the burden. They didn't come close to meeting it on the
6 various factors.

7 Analyzing the decisions, many defendants claim:
8 We're doing you a favor. We're helping you out. You don't
9 realize it. You don't like it. So you're suing us, but
10 actually we're helping you. Court's don't give that much
11 weight.

12 **THE COURT:** Here Mr. Schwartz said that rockets were
13 being put on Java. I mean, that -- that's a very helpful
14 document for Google here.

15 **MR. JACOBS:** One week later Rich Green, senior
16 executive at Sun, is published in an article that Google said
17 it saw in which he said, "We're very concerned about the
18 fragmentation of Android." Rockets on Java is before the SDK
19 is released. Rich Green's comment about fragmentation is after
20 the SDK is released.

21 Jonathan Schwartz, no friend --

22 **THE COURT:** Why is there more fragment -- didn't
23 Google have the right to -- let's say that Google had written
24 using the Java language it's own set of APIs -- it didn't even
25 use the same words, names, whatever -- from the ground up, but

1 it used the Java language. So it's out there.

2 And, surely, you would admit that they had the right
3 to do that; correct?

4 **MR. JACOBS:** Separating out the APIs, just
5 implementing the --

6 **THE COURT:** Oh, no, no, no. They have their own
7 APIs. They don't even use the same names. They have got them
8 organized differently. There is no SSO problem. It's a
9 completely different SSO, completely different set.

10 It has the same functionality spread around just
11 like -- you can't possibly claim you have the right to ask a
12 method to tell you the cosign of an angle.

13 **MR. JACOBS:** That's the Spring case came in, the
14 evidence on Spring.

15 Spring is a company, as the testimony undisputed
16 revealed, that has implemented an entirely different set of
17 APIs, but supports the Java programming language. We know how
18 Google supports the Java programming language --

19 **THE COURT:** All right. Let's say Spring -- are you
20 saying that Spring did something wrong?

21 **MR. JACOBS:** No. Spring did not implement the API
22 specifications owned by Oracle.

23 **THE COURT:** If Google could have done what Spring
24 did, why is there any greater -- there there would have been
25 immense fragmentation. It would have been completely

1 different. It would have been a Java-based platform that had
2 nothing to do with your version of Java, but instead they did
3 one that was part -- 37 were consistent. The others were not
4 consistent. And why -- if they could do the greater, why
5 couldn't they do the lesser?

6 **MR. JACOBS:** Because the lesser has its own
7 commercial burdens. They would have had to make their own
8 investment. They would have had to make their own investment
9 in training developers. They would have had to make their own
10 investment in creating the APIs in the first place.

11 I mean, that's the nature of intellectual property
12 protection, your Honor.

13 **THE COURT:** I understand possibly that point, but I'm
14 talking about in terms of fragmentation.

15 **MR. JACOBS:** Fragmentation is defined in the Java
16 environment as implement -- subsetting or supersetting the API
17 specifications. That is what the Java Community understands
18 fragmentation to be. Again, that was undisputed.

19 And so if they want to do something that creates a
20 whole new nother world, we can't stop that and the Java
21 Community wouldn't understand that as fragmentation either.
22 They would understand that as Google using the Java programming
23 language --

24 **THE COURT:** There is no decision anywhere that says
25 copyright prohibits somebody from supersetting, let's say, one

1 method -- one class.

2 **MR. JACOBS:** Of course not, your Honor.

3 **THE COURT:** All right. So that's just -- that's just
4 the business model.

5 All right. So I guess your argument comes down to
6 saying: We didn't want anyone to superset our classes and that
7 hurts us in some way. And you call that fragmentation.

8 **MR. JACOBS:** There are internal emails at Google
9 acknowledging fragmentation concern. And you heard a lot of
10 testimony about how Google enforces anti-fragmentation in its
11 world through anti-fragmentation provisions, through testing
12 suites. It's an entirely analogous model. And you heard now
13 in Phase two they have invested tremendously in
14 anti-fragmentation.

15 So, fragmentation is a concern for any platform
16 developer who is trying to create an ecosystem and doesn't have
17 a closed model like -- say, like Apple. If you want to
18 discourage that kind of model, if you want to discourage open
19 models in which independent implementations are allowed, but
20 there's a set of licensing restrictions and agreements that
21 create consistency between those platforms, then go with Google
22 on fair use because it would be devastating to the Java model
23 if people can pick and choose at will and fragment to their
24 heart's content. Because if Google can do it, so can the next
25 guy. And it may not be so --

1 **THE COURT:** But aren't you the one that wanted the
2 jury in this case? Didn't I hear Mr. Van Nest at one point say
3 they would waive a jury, and you said no, you wanted a jury.
4 So now we have the jury's work on this.

5 **MR. JACOBS:** We do, your Honor.

6 **THE COURT:** But now you don't like what the jury came
7 out with. You want the judge to make a ruling.

8 **MR. JACOBS:** You're right.

9 **THE COURT:** Well, I'm not sure you're entitled to it.

10 **MR. JACOBS:** That's our burden to show you that we
11 are, your Honor.

12 **THE COURT:** I have a different question I want to ask
13 both of you to address. No one has addressed this and maybe
14 it's because it's just completely off base.

15 It's going to take a minute to develop this, because
16 we have three, three of -- at least three and maybe four of
17 these packages are referred to as core.

18 The original -- when the language first came out, the
19 book was all -- the book on the language, and it included the
20 three. I think it was IO -- java.lang, java.io and
21 java.something else.

22 **MR. KWUN:** Java.util.

23 **THE COURT:** Util, yes. And this had things that in
24 other languages are just part of the language, like return the
25 cosign. Return the tangent. Return the greater of two things.

1 Return the absolute number. Those are things that in other
2 languages are just one of the normal parts of the language.
3 And these were all lumped together as -- or print, print
4 function. Now, at that time no sharp distinction was made. No
5 distinction was really made between the packages and the rest
6 of the language.

7 As time went on, the programming people who liked
8 Java could see that it was a handy way to do pre-packaged
9 programs that would do things a lot harder than return the
10 cosign or the tangent, and this -- this group of 37, and now
11 166, packages grew up that had many, many, many functions. 37
12 of those were duplicated in some sense in the Android.

13 Now, one of the reasons I had broken out in that form
14 of -- special verdict form that got rejected by both sides was
15 that I thought it was plausible that a jury could say those
16 first three -- util, IO, and lang -- it was fair use to use
17 those three because the language wasn't any good without it.
18 But because of the ownership issue, Oracle withdrew a
19 one-by-one package analysis and went as a group, okay? So I
20 said, Fine, we'll go with it the way you want to go with it.

21 But it concerns me that you would be asking for a
22 home run on a question we put to the jury on where, to my mind,
23 at least on those three there is a very strong argument for
24 fair use, which in and of itself would deny a global win for
25 Oracle on this point. And Oracle is the one that chose to put

1 it to the jury in that way for reasons that have to do with all
2 of that ownership stuff that we spent many afternoons debating.

3 I don't know how to -- you know, this is -- this is a
4 complication that I need some guidance on from you, but it
5 weighs on my mind. I see some of -- I see some of the strength
6 of the arguments made by Oracle on these factors, but it would
7 be to my mind wrong to allow those three packages to be -- I
8 think the fair use argument there is very strong. Certainly, a
9 jury could say that it was fair use to use those three and
10 based on that alone would have been entitled to say, no -- I'm
11 sorry, reject Oracle's view.

12 Now, that's the one on which, though, the burden of
13 proof was on Google. So even if they could prove three, do
14 they have to prove -- do they have to prove all 37? Do they
15 just have to prove one?

16 We didn't get into that level of detail and it didn't
17 occur to me until after the verdict that that was lurking
18 there. And maybe you all saw that and just let it go, you
19 didn't want to raise it. But that's where we are now.

20 So I'd like to get your views on the fair use issue
21 as it applies to those three. Maybe there's something I don't
22 see.

23 And here is another complication. You didn't make on
24 the Google -- you did not make a JMOL on that issue. No Rule
25 50 by Google on these three packages. And maybe you played the

1 hand that way for precisely the flip side reason that you were
2 going for the home run yourself.

3 So I don't know where this -- you know, I have been
4 thinking about this case, you know, in my own way. I have been
5 trying to work my way through it. And I see those three as
6 possibly very different than the rest of this case, and so I'd
7 like to hear your views on that subject.

8 Mr. Jacobs.

9 **MR. JACOBS:** I think there are two questions lurking
10 there. One is the fair use merits of those three packages, and
11 the second is how the possibility that there might be a
12 dividing line between those three and the rest analytically
13 might effect the JMOL motion.

14 On the first, we have to be clear what we're talking
15 about. We're talking about packages that were referred to in
16 the language specification, but not fully specified, and
17 certainly not fully specified in their current form.

18 So if the idea was Oracle/Sun made some declarative
19 statement about the programming language being free for all, we
20 tried to figure out what that declarative statement means and
21 its practical impact on packages. And so we go back to 1996
22 and we look at what the programming language declaration might
23 have said then.

24 Now, keep in mind the evidence on this is not very
25 clear in terms of what was actually said and what was actually

1 made available for use by all, but let's stay with the
2 hypothetical anyway.

3 There's a declaration made in 1996. The programming
4 language is free for all. There is a book that you can look
5 at. And that which is -- so, therefore, one thinks that that
6 which is specified in the book that goes beyond the formal
7 definition of the programming language is available for all.
8 That is fragments. That is fragments of what we're talking
9 about today with the 37 packages.

10 So whatever the fair use merits -- and I think I
11 would like to get to the second question first.

12 If somebody takes a copyrighted work and copies it
13 and it turns out that during the course of the litigation some
14 component of the copying is not justiciable or is not
15 probative, maybe it's held to be uncopyrightable, that doesn't
16 mean that the infringing -- that you divided up the judgment.
17 Absent some proper motion by the defendant, that would result
18 in -- would result in that.

19 So take the *Abden* case again. There were plot
20 elements in *Rear Window* that were common place, scenes a faire,
21 et cetera. You could show that all you want, but the replay of
22 *Rear Window* in *Abden* is a copyright violation.

23 So I don't think the possibility, kind of
24 unadjudicated and not fully litigated possibility of fair use
25 analysis on these packages should interfere with JMOL,

1 particularly as we settled on the definition of the work as a
2 whole. Because the work as a whole here, I guess, on the
3 giving side, if you will, on the copyright holder side is the
4 166 packages, and the accused packages are the 37 Android
5 packages taken as a group.

6 I don't think they get off the hook if eency-weency
7 bits of those packages would be held to be non-infringing under
8 any theory.

9 Thank you.

10 **THE COURT:** Thank you.

11 Mr. Baber.

12 **MR. BABER:** Your Honor, just a few comments on the
13 issue of the core packages, as opposed to the others.

14 Obviously, our -- we also have the issue of whether
15 or not those are copyrightable, the whole issue that's out
16 there. That clearly --

17 **THE COURT:** That's a separate point. If you were to
18 win on that, then this will all be moot. But we're assuming
19 right now that you lose on that.

20 **MR. BABER:** That's right.

21 The reason why we didn't urge a separate jury verdict
22 question just on some or all of those packages is because
23 Oracle had withdrawn the claim for findings of infringement as
24 to those packages. And the way the verdict was set up, you had
25 to first find infringement and only if you found the

1 infringement as to something, whether it was the SSO, whether
2 it was the documentation, only then would you reach a fair use
3 issue.

4 So that -- I think in terms of the process and
5 procedure and how we got there, that's why nothing got broken
6 out package-by-package, whether it was their claim of
7 infringement or our fair use defense.

8 But, in fact, your Honor, the record contains
9 evidence. There's two parts to the evidence as to how these 37
10 packages relate to the language.

11 You had testimony from Dr. Bloch and Dr. Reinhold of
12 Oracle, who both agreed that there are some 60 or 61 classes
13 within the 37 packages that are necessary to use the language.
14 Doctor --

15 **THE COURT:** Those are only in those three packages I
16 mentioned?

17 **MR. BABER:** I believe they are scattered in those
18 three packages that you mentioned, your Honor.

19 Dr. Bloch then gave a second level of analysis and he
20 said, but in order to fully implement those 61, you need things
21 from a bunch of other classes. And it wound up being on the
22 order of 2,000 different methods and fields, et cetera.

23 **THE COURT:** Are those all still within those three
24 packages?

25 **MR. BABER:** No. Those then expand to about 10 of the

1 37 packages, as I recall.

2 But then you had testimony at the very end --

3 **THE COURT:** Is that really in the record? What are
4 the names of those 10?

5 **MR. BABER:** They are not in the record, your Honor,
6 and we didn't do it one-by-one, just like they didn't do their
7 packages one-by-one, because I think what wraps it all together
8 is at the end of our case, Professor Astrachan was on the
9 stand, and Professor Astrachan first said, yes, I agree with
10 Dr. Bloch's analysis, both the first level and the second
11 level, and I agree with Dr. Reinhold's analysis.

12 And Dr. Reinhold also did analysis, you may recall,
13 of which classes were necessary, had to be known to the
14 compiler. So he came up with a different number, I think 40 or
15 so.

16 But then Professor Astrachan went a step further, and
17 he said, Well, there are clearly these parts of these packages
18 that are necessary to practice the language, but all 37 of the
19 packages that are in Android are necessary as a practical
20 matter.

21 **THE COURT:** That's a different point. That's
22 different. I mean, it's one thing to say the tentacles of
23 the -- what was it -- 61 classes reach out to another 100
24 classes and they are spread over 8 or 10 packages, and so you
25 take those and you -- somehow you put that in group one.

1 But let's say it's 10. That still leaves 27 that
2 are -- the tentacles don't even touch. So that -- that only --
3 you have to fall back on your -- the developers are expecting
4 these to be there as a practical matter.

5 **MR. BABER:** That's correct, your Honor. That's why I
6 distinguished --

7 **THE COURT:** That's a different kettle of fish.

8 **MR. BABER:** Correct. And, again, a lot of that,
9 frankly, that was in the record, your Honor, was for you on
10 your issues of copyrightability. And --

11 **THE COURT:** So, all right. All right.

12 **MR. BABER:** And then I would also observe, just to
13 the point that Mr. Jacobs just made, which is, well, if there's
14 parts that are maybe scenes a fair, et cetera, that goes back
15 to the whole methodology for determining infringement, which is
16 anything that's not copyrightable, whether it's because it's a
17 Section 102(b) method of operation, whether it's because it's a
18 functional requirement for compatibility, whether because it's
19 a scenes a faire, those are supposed to be removed from
20 consideration before the analysis for copyright infringement.

21 You go through the process of what the Ninth Circuit
22 calls analytic dissection or what the Second Circuit calls, you
23 know, abstraction filtration comparison.

24 At the end of the day, all you're supposed to compare
25 are the parts that are copyrightable. And given your Honor's

1 instruction to the jury, you should assume, you should take it
2 for granted, that this is copyrightable. Getting down to this
3 level of granularity as to whether one class or another class
4 in java.io was necessary for the language, frankly, would have
5 been an impossible task for the jury given the assumption that
6 these are copyrightable for purposes of their analysis.

7 **THE COURT:** All right. We need to move on.

8 And now we'll go to a -- we'll go to your motions, so
9 we'll take one from each side. So we've done one from the
10 Oracle side, and we'll do one on the JMOL side for Google.

11 And the one that I think is the one that I would like
12 to hear the most about is your one that -- and only one part of
13 it, and that is that your view that declarations are not
14 copyrightable. I guess as a matter of law you're saying that.

15 So I would like -- and I would like as you make this
16 argument, for you to be very precise on what you mean by
17 "declaration."

18 All right. Go ahead.

19 **MR. BABER:** Yes, your Honor.

20 What we mean by the declarations in our JMOL motion
21 is what's been referred to at trial as the method signatures
22 for the methods and the fully qualified names, Dr. Bloch talked
23 about.

24 **THE COURT:** All right. Would you -- do we still have
25 that chart?

1 And you show me what you mean because the books that
2 you put in evidence use the word "declaration," I think,
3 differently.

4 Can you bring it a little closer? That's good right
5 there. Thank you.

6 (Demonstrative displayed)

7 **MR. BABER:** On your last point, your Honor, I believe
8 that's correct and I believe Professor Astrachan mentioned that
9 in his testimony; that in the language book when it uses the
10 phrase "declaration," it would include all the implementing
11 code as well.

12 **THE COURT:** Correct.

13 **MR. BABER:** Okay. But the way all the witnesses
14 testified at trial, I believe -- and it's on Dr. Bloch's
15 chart -- this is what everyone has referred to as the
16 declaration (indicating).

17 **THE COURT:** Certainly, he did. And maybe all of them
18 did, but I can't tell you whether all of them did.

19 But you mean the part that's in green.

20 **MR. BABER:** Well, it's inside the black box.

21 **THE COURT:** Is that black or green?

22 **MR. BABER:** It's black.

23 **THE COURT:** Black, okay.

24 **MR. BABER:** The fully qualified name for this is
25 what's in the green boxes. `Java.lang.Math.max`, that's the

1 fully qualified name, but this is the declaration of the
2 methods.

3 **THE COURT:** Just so it's clear for the record, it
4 says "public static" -- I will do it exactly.

5 Public space static space int space max. No space.
6 Paren. Space -- no, no. Would there be a space there?

7 **MR. BABER:** No, it would be inside the paren.

8 **THE COURT:** So int space arg1 comma space int space
9 arg2 close paren, end of declaration.

10 **MR. BABER:** Correct.

11 **THE COURT:** All right. And what is your argument
12 there?

13 **MR. BABER:** The reason we put this in our JMOL
14 motion, your Honor, is you already ruled on summary judgment
15 that the package names, java.lang, the class names,
16 java.lang.Math, and method name java.lang.Math.max are
17 unprotectable as words and short phrases under copyright law.

18 We believe the same is true of the declarations.
19 They are short phrases and we believe they are not separately
20 and individually copyrightable.

21 If you look -- it's just a question of clarification
22 as to how far the ruling you already made on short names and
23 phrases goes. We think the length of these method signatures
24 or declarations are similar in length to the short bit --

25 **THE COURT:** Let me give you a different argument that

1 works in your favor on that very point. And the reason I want
2 to do this is because I'm thinking about it and I want to give
3 Mr. Jacobs a chance to shoot it down.

4 I'm not saying this is what I'm going to rule. I'm
5 just saying this is what I'm thinking.

6 Now, I want to take my cosign example -- well let's
7 take this example. Let's take this example. This is just as
8 good.

9 Java has an API. Within the API are 37-plus
10 packages. One of those packages is java.lang. It has a class
11 called Math. Within Math there is a method. Actually, several
12 methods that get the maximum of two numbers. This one that we
13 have on the board is -- uses integers, if I understand it
14 right. So it's only for the case where you have whole numbers
15 as opposed to fractions.

16 So here is -- that's background.

17 Now, that is a concept or function that -- there
18 would be many ways to write the implementation. We saw four
19 ways during the trial. You could probably come up with other
20 ways to write it.

21 I think Oracle would concede that it cannot claim a
22 copyright over the idea of a method that would take two numbers
23 and return the bigger of the two numbers. Just like it could
24 not possibly claim to have a copyright over any and all ways to
25 take a number, an angle, and return the co-sign merely because

1 it has a copyright on its way to do that.

2 So the public, the universe, is free to come up with
3 its own method for comparing two numbers and returning the
4 larger, so long as it does not use the specific code developed
5 by Java.

6 Now, that line, though, is going to have to be the
7 same, the declaration. If you want to have that function
8 carried out, then under the rules that the programming language
9 imposes on the user, you've got to use the word "public" if you
10 want that function. There's the word "private." There's like
11 a -- you know, there are several choices on each one of those
12 words. There's "public," "private." And then on static
13 there's several possibilities there. I think the word "void"
14 is one, maybe. Integer you can have "double" instead of
15 integer.

16 The word "max" is a name. I've already ruled that
17 the word "max" is not protectable.

18 You can vary the arg1. You can put an "x" and "y."
19 That part can vary, right?

20 **MR. BABER:** Your Honor, absolutely correct, Your
21 Honor.

22 **THE COURT:** All right. So the proposition I put --
23 and this is really a question to Oracle -- isn't that one line
24 controlled by the merger doctrine?

25 There's only one way -- if you want to have that

1 function -- and everyone has the god-given right to have that
2 function. Oracle does not have a monopoly on it.

3 If you want to have that function, that is the only
4 way to write it. Therefore, merger would protect the right of
5 the public to use that form of declaration.

6 Now, I want to pause here and say, that would only
7 help Google insofar as at the method level. That does not --
8 that would not be an answer to, Why did it happen to be that
9 your methods got put into the same classes?

10 You could have had exactly that same method and put
11 it under the IO. You could have had it under the IO. You
12 could have had it under any of those other classes or packages.
13 But you mimicked exactly the same -- but just take the
14 declaration level for a method. I put to you all the
15 proposition that the merger doctrine would protect that.

16 So why don't you have a seat and let me hear what
17 Mr. Jacobs has to say, because I think this is something I did
18 not understand going into the trial, but I will say this is the
19 way I'm leaning.

20 You can kind of tell from the way I'm talking I've
21 thought about it, and this is the way I'm leaning on that one
22 line. For every single method, this is going to be the same
23 analysis.

24 Go ahead.

25 **MR. JACOBS:** Well, our case is not about any single

1 method or any group of methods.

2 And the wisdom in the Court's instruction was to link
3 the protectability of names or, in this case, method
4 declarations, as we're now using the terminology, to the
5 structure, sequence, and organization of the software.

6 And this is the key distinction. In any copyright
7 case, you could get very granular and you could say, well,
8 that's an unprotectable idea. That's an unprotectable idea.

9 But, of course, the plaintiff isn't suing for copying
10 this idea or that idea. In any copyright case of any gravity,
11 the plaintiff is suing for some combination of elements, any
12 one of which could be characterized at the idea level. But
13 because of the way they are combined, they represent original
14 expression.

15 In a music case, no note is protectable. And
16 probably no diad of notes. But you get to five notes, six
17 notes, seven notes, all the sudden you have protectable
18 expression.

19 So is this case about the max method? Max is a
20 trivial method, so it's probably an unfair example --

21 **THE COURT:** But this principle is throughout because
22 every class has many methods. And every package has many
23 classes.

24 So the method thing is going to be there, I'm
25 guessing, several hundred times in the overall problem we have

1 before us.

2 And you know those cases do say that what I'm
3 supposed to do is -- is wade through this, in excruciating
4 pain, to find the part that is protectable and the part that's
5 not protectable. And then with the part that is protectable,
6 to then that's the part you ask about fair use on.

7 So, you know, now that I've heard all this evidence,
8 that's what I'm trying to do, is to -- so I did say something
9 close to what you just said, which is, okay, even if this is
10 right, does that then explain away the -- does the merger
11 doctrine explain away why it happens to be that all of those
12 methods are lockstep found in the Google version of the math
13 class, for example? And it does not. It does not.

14 Now, maybe something else would. For all of you this
15 may have been already obvious. But for me it didn't start to
16 dawn on me until I tried to understand what this -- all these
17 words mean.

18 And I think I understand the declaration level. All
19 right. So maybe you're agreeing with me up to the point of
20 the -- sounds like you're agreeing with me up to the level of
21 the method declaration.

22 **MR. JACOBS:** Well, I'm not -- I think I'm agreeing
23 that -- what I'm not trying to do is argue over max.

24 We had in our brief some examples of method
25 declarations that were several lines long and are not as

1 trivial as max. So my proposition to you is that max was a
2 useful teaching device in part because it was so simple we
3 could get it all on one page, including what Google
4 characterizes as the implementing code below.

5 But if you examine other method declarations --

6 **THE COURT:** No, I've been looking at some of these.
7 I agree with you. They could be many, many, many lines long.

8 But isn't this still true? Every single word in a
9 declaration serves a functional purpose.

10 **MR. JACOBS:** Every single line of code in a --
11 probably not. But it doesn't matter. Every single line of
12 code, of executable code in a computer program, serves a
13 functional purpose.

14 **THE COURT:** Yes. I didn't say it quite right.

15 You don't have the right, the ownership, of every
16 single way to do every single method.

17 Anybody has the right to mimic -- let's say you came
18 up with a great way -- I'll use this example. This is not so
19 trivial. Let's say you wanted to have a method that would take
20 the month and the day, and maybe even the time, and figure out
21 what the declination of the earth was to the sun.

22 So you wrote a method that would have to be more than
23 one line long. It might be, I don't know, 20 lines long. And
24 let's say that you chose all those publics and the statistics
25 and whichever way you wanted to do it.

1 You wouldn't have the right to say: Okay, we've now
2 discovered how we can do this. We now own this under copyright
3 law. No one else can come along and do the exact same
4 specification.

5 I don't think you have the right to say that.

6 **MR. JACOBS:** But it's really -- with respect, Your
7 Honor, it's a false hypothetical for our case.

8 **THE COURT:** Well, it helps me to -- why is that?

9 **MR. JACOBS:** Because what we have here is a case of
10 the comprehensive taking of the entire structure, sequence and
11 organization.

12 **THE COURT:** I can see that's a different issue.

13 **MR. JACOBS:** And what we see in the copyright
14 cases -- I mean, look, this is copyright, and it's being
15 applied to computer software so we're struggling with it.

16 But, nonetheless, copyright law is pretty clear on
17 this point. If you get over-granular and say, you know, this
18 name in the phone directory, you can't be the only one to have
19 a phone directory over that name. That name is not
20 protectable. But, you know, creating a business directory of
21 phone listings is protectable.

22 This is blurring into an originality issue as opposed
23 to a copyright --

24 **THE COURT:** I'm not saying originality. No, no, no.

25 I'm saying if you want -- it's the merger doctrine.

1 If you want to have a way to specify, this is what we're going
2 to put in, this is what we're going to put out, and you want it
3 to be public or private, or whatever, that is a function. And
4 you say -- and you -- once you decide how you want it to
5 unfold, then there is a precise way to use the declaration to
6 say it. And the fact is, there's only one way --

7 **MR. JACOBS:** No.

8 **THE COURT:** -- it can possibly be said.

9 **MR. JACOBS:** And that's factually wrong. And it's
10 testable.

11 **THE COURT:** I don't believe that. Okay. Explain why
12 I'm wrong.

13 **MR. JACOBS:** It's testable. Let me start with it's
14 testable.

15 **THE COURT:** Okay. Tell me why I'm wrong.

16 **MR. JACOBS:** Because Google could have looked at
17 Application Programming Interfaces and method declarations.
18 They could have said, look, there is no claim that we copied
19 the SSO of the non-37 packages. But, look, lo and behold, when
20 we were doing the same method purpose as was being done in
21 Java, in our own independently-created, you know, 38th, 39th
22 API, lo and behold, we came up with the same method
23 declaration.

24 And there's no claim of copying there. Proof: Two
25 programmers doing the same task would come up with the same

1 thing.

2 They never introduced any evidence like that.

3 **THE COURT:** They don't have to. I'm willing to
4 assume they copied that part. And I'm saying to you that the
5 law would protect them, that you don't have the right to
6 monopolize that method.

7 **MR. JACOBS:** We don't have a right to monopolize a
8 collection -- we don't have a right to monopolize the ability
9 to carry out this function by monopolizing the words associated
10 with that function. I think --

11 **THE COURT:** But those words in that box can only be
12 said that one way.

13 **MR. JACOBS:** We already saw that the variables could
14 be different.

15 **THE COURT:** And, in fact, they are different, aren't
16 they, in our case? They didn't copy those; did they?

17 **MR. JACOBS:** No, no. We have lots of evidence of
18 direct copying of variables that could be different. I think
19 hundreds and hundreds of them.

20 **MR. KUWAYTI:** Two-thirds --

21 **MR. JACOBS:** Two-thirds of them, Your Honor. They
22 were counted. And they are in the record.

23 But, again, this is trivial. The relevant question
24 is that if you gave a programmer an assignment, carry out this
25 purpose, would there be only one way to write a declaration to

1 do that? Or very few ways, such that they really all look
2 similar?

3 That's the merger doctrine. That's experimentally
4 provable, and Google --

5 **THE COURT:** That's true. I'm saying there's only one
6 way to do it.

7 **MR. JACOBS:** And this --

8 **THE COURT:** That's why they have the right to have --
9 they have the right to have their own implementation. And
10 merely because it is -- the declaration has to be written in
11 one way to get there, that doesn't block this like -- you're
12 saying you've got a monopoly over all ways to do it.

13 **MR. JACOBS:** Again, all I'm saying is that's an
14 empirical or a fact-driven question as to declarations in
15 question, and that you'd have to have proof on the topic, not
16 just surmise based on examination. And we have only this
17 example (indicating) --

18 **THE COURT:** No.

19 **MR. JACOBS:** And Google never argued that this was
20 representative.

21 **THE COURT:** I've looked at those rules and those
22 books you gave me, and put in -- the word "public" has very
23 precise meaning. The word "static" has a very precise meaning.
24 The word "int" has a very precise meaning. All of it.

25 And the word max: is a name. And that's not

1 protectable. They can borrow that name all they want.

2 **MR. JACOBS:** So, then, look at the declaration at the
3 bottom of page 12 of our brief: Public abstract void verified
4 public key key string sig provider throw certificate exception
5 no such algorithm exception invalid key exception no such
6 provider exception signature exception.

7 That's a method declaration. There are probably a
8 lot of ways to write that method declaration. If it were true
9 that for even the majority of the method declarations in Java
10 there were only one way to do it, Google could have proven
11 that.

12 **THE COURT:** I think it's in the rules. I think it's
13 in the rules of the language that if you want to have an
14 overall method that does what that declaration specifies, that
15 is the exact and only way to do it, with the exception of you
16 could have said X and Y instead of arg1 and 2.

17 **MR. JACOBS:** So in the hypothetical that I gave, in
18 the hypothetical, the real one -- this is in
19 java.security.cert.certificate -- "public" is defined by the
20 language. I think. I'll check. "Abstract" is defined by
21 the --

22 **THE COURT:** That's another word that comes in the
23 language. I've seen that in reading. Abstract is one of the
24 keywords.

25 **MR. JACOBS:** And I believe "void" is also a language

1 construct, and maybe even verified. But --

2 **THE COURT:** It means that no -- there's no return.
3 "Void" means there is no return.

4 **MR. JACOBS:** But then the rest of that method
5 declaration, public key key string sig provider throw
6 certificate, et cetera, is subject of many different ways to
7 write it.

8 We heard a lot from Mr. Bloch about the creativity in
9 selecting and choosing and writing and authoring the right
10 names of these Application Program Interface constructs.

11 **THE COURT:** I've already said names -- you know, the
12 Federal Circuit may reverse me on this, but, in my judgment,
13 names are not protectable.

14 **MR. JACOBS:** And we --

15 **THE COURT:** No matter how long they are, they are not
16 protectable. They can use those names all they want.

17 **MR. JACOBS:** And we are not challenging, in this
18 argument, that conclusion. We are agreeing with your -- with
19 your instruction that while individual names are not
20 protectable on a standalone basis, names must necessarily be
21 used as part of the structure, sequence and organization, and
22 are to that extent protected by copyright. This is not a case
23 of --

24 **THE COURT:** Well, that's what I told the jury, and
25 that -- that's correct. That's what I told the jury.

1 But this whole thing of giving it to the jury was on
2 the assumption -- what I was trying to do was to avoid having
3 to retry the case, so we could just have one trial.

4 But that -- I'm not even sure that what I told the
5 jury was actually correct as a matter of law on -- that as part
6 of the SSO they are somehow not protected.

7 **MR. JACOBS:** So let me just recapitulate what we
8 think the strongest argument to you on this point is.

9 Our case is not about the taking of any individual or
10 even any small set of method declarations.

11 Our case is about the comprehensive taking of -- to
12 use the language of the instruction -- the structure, sequence
13 and organization of the computer programs as defined by the
14 Application Programming Interface specifications.

15 That structure, sequence and organization includes
16 method declarations at the appropriate level. It is like the
17 sub sub subchapter in the outline structure.

18 The code down here (indicating), if you're a
19 Microsoft Word person, this is body text. And this is in the
20 outline.

21 And what we are seeking to protect is our very
22 complex outline. It would not be a relevant question, if you
23 were protecting a particular taxonomy, whether any particular
24 element of the taxonomy -- whether plants from Bulgaria is
25 protectable, the relevant question would be: Did the defendant

1 take the entire outline structure of a book on plants from
2 around the world, in which plants from Bulgaria was one tiny
3 fragment of what was taken?

4 No one would bother to ask the question whether
5 plants from Bulgaria was taken because that's not our claim.

6 **THE COURT:** All right. Let me ask Mr. Baber, to
7 respond.

8 Let's assume that the Court is right on what I said a
9 moment ago. That still does not answer most of what Mr. Jacobs
10 just said, in a way.

11 In other words, even if the method has to be written
12 in the way it's put there, that method could have shown up
13 under the input/output. It doesn't have to be even in the
14 package or the class.

15 You could have put it anywhere you wanted, and still
16 had the same functionality. And the problem would have been
17 that the developer community would not have -- would not have
18 liked that. They would have said, Why did you put max in the
19 wrong place? That's what they might have said.

20 So the fact -- that's just a business thing. That's
21 not required by the language itself. There is more than one
22 way to organize the SSO in the broader -- in the package and
23 class level.

24 So even if it's true that this declaration is --
25 that's the only way to write the declaration, so merger

1 protects it, that doesn't answer the whole SSO problem.

2 **MR. BABER:** One step at a time, Your Honor.

3 **THE COURT:** Mr. Jacobs didn't even like me taking
4 that step. But I'm thinking about that step.

5 But having thought about that step, I see it as just
6 like one step on a five- to six-step process. And you still
7 have a long way to go.

8 **MR. BABER:** Well, Your Honor, take it one step at a
9 time. First step is, I think you're absolutely correct on that
10 issue that the form of a method declaration is the epitome of
11 the merger doctrine.

12 **THE COURT:** You didn't have to use arg1 and 2.

13 **MR. BABER:** No.

14 **THE COURT:** Is it true that two-thirds of the time
15 you copied even that?

16 **MR. BABER:** I don't recall candidly, Your Honor, who
17 testified about that or what they said. But there are some
18 common variables that are generally used, X and Y --

19 **THE COURT:** X and Y, A and B, I and J. You know, X1,
20 X2. But there's more than one way to write it.

21 **MR. BABER:** But just to get more granular, if what
22 you want to have is a public method, something developers can
23 access, that they can call on and invoke it, it's a static
24 method that returns an integer --

25 **THE COURT:** Remind me what "static" means.

1 **MR. BABER:** I'm going to defer --

2 **THE COURT:** Mr. Baber, you're shocking me.

3 (Laughter)

4 **MR. BABER:** I'm sorry, Your Honor. I'm still
5 learning this stuff as I go. One of our folks will have to
6 explain exactly what "static" means, as opposed to "void" or
7 the other words that can go in this space.

8 But if you want to have a public method that is
9 static, that returns an integer, and it takes as its input two
10 integers, this is the only way you can write that declaration
11 consistent with the language specification.

12 **THE COURT:** I think I agree with that. I think I
13 agree with that. And the only parts that you would have any
14 flexibility on are the name and what to call two variables. I
15 think, otherwise, it's dictated by the rules of the program.

16 **MR. BABER:** And I think that same --

17 **THE COURT:** Rules of the language.

18 **MR. BABER:** I agree, Your Honor.

19 And I think that's true no matter how simple or
20 complicated the method is. The example Mr. Jacobs gave, again,
21 if you want to have a public method that returns --

22 **THE COURT:** All right. All right. So how do you
23 address the more fundamental point? Okay. Let's say you win
24 on step one. How do you get all the way to the package level?

25 Because you do have exactly the same lineup and

1 outline and taxonomy as the -- so how do you explain that part?

2 **MR. BABER:** Your Honor, that is driven a hundred
3 percent by the language requirement for fully qualified names.

4 You've already ruled we have the right to use the
5 names at each level. Package name. Class name. Method name.

6 And you asked a question this morning --

7 **THE COURT:** Let's assume that's right. You could
8 have put this -- you could have put max not under the Math
9 class. You could have put it under a different class.

10 **MR. BABER:** You could have. But then you get -- you
11 move from merger to a different copyrightability issue, which
12 is functional requirements for compatibility. Which is,
13 someone who's used to these API methods, who's used to calling
14 max all the time, they know it's java.lang.math.max. And in
15 order for their code that they've written in the past to work,
16 in order for them to continue to use the API methods they've
17 memorized for compatibility reasons -- Professor Astrachan
18 talked about this both on the part of developers as well as on
19 the part of the part of industry --

20 **THE COURT:** Is this a fair use argument or
21 copyrightability issue?

22 **MR. BABER:** This is a copyrightability issue, Your
23 Honor.

24 **THE COURT:** Where does it say that in the law, that
25 protectability turns on this compatibility idea?

1 **MR. BABER:** *Sega vs. Accolade*. The Ninth Circuit
2 said that, quote, functional requirements for compatibility are
3 not protected under Section 102(b). It's an idea method system
4 point.

5 **THE COURT:** That's -- okay. That is the -- you know,
6 that is the big, big issue in the case, is 102(b). So you fall
7 back on the atomic bomb, the nuclear --

8 **MR. BABER:** No, I've got some other bombs, too.

9 **THE COURT:** 102(b) is a nuclear option. That's the
10 big issue.

11 Maybe you're right about that, but I'm searching for,
12 is there a way to get there without getting to 102(b).

13 **MR. BABER:** Yes, you can get there on merger as to
14 the class level as well.

15 **THE COURT:** No, not at the class level. Because you
16 could have put that -- in more than one class, you could have
17 put that max method in.

18 **MR. BABER:** We could have, Your Honor. But we
19 believe under your prior rulings we have the right to use the
20 fully qualified, name `java.lang.math.max`, which puts it in that
21 class.

22 **THE COURT:** If that's what I ruled -- I don't think I
23 did. I thought I held off on that.

24 But if that were true, yes, you're right, because of
25 the rules of the -- you would have had the right to -- that's

1 the only -- that's right. I don't think I said that. I think
2 I -- I thought I said for multi-word names we were going to
3 hold off.

4 **MR. BABER:** Just a second, Your Honor.

5 **THE COURT:** Well --

6 **MR. BABER:** Sorry. I don't have it in this. I had
7 it in the brief --

8 **THE COURT:** My memory could be wrong, but I thought I
9 said for the longer names I wasn't sure. I wouldn't say you
10 were wrong on that. I just said I wanted to have the trial
11 first.

12 **MR. BABER:** In your summary judgment order, Your
13 Honor, you say you find that the names of the various items
14 appearing in the disputed API package specifications are not
15 protected by copyright.

16 I was just trying to see --

17 **THE COURT:** I'll go back and read it and see. But
18 the implication of your argument to say that then
19 java.lang.math.max is protected, that that -- that destroys the
20 SSO argument right there.

21 **MR. BABER:** I found it, Your Honor.

22 **THE COURT:** All right.

23 **MR. BABER:** Summary judgment order, page 7. You talk
24 about the API package specifications. You say, quote: Words
25 and short phrases such as names, titles, and slogans, unquote,

1 are, quote, not subject to copyright, unquote.

2 You cite regulation 202.1. You cite the *Planesi*
3 Ninth Circuit opinion.

4 "Google argues that, quote, the names of the
5 Java Language API files, packages, classes,
6 and methods are not protectable as a matter
7 of law." Closed quote. Cite to our brief.

8 "This order agrees."

9 Because names and others -- sorry. Lost the page.

10 "Because names and other source phrases are
11 not subject to copyright. The names of the
12 various items appearing in the disputed API
13 package specifications are not protected."

14 **THE COURT:** Well, what was the part that I said --
15 there was more to it than that. There was something I said I
16 was going to wait until the trial was over before I decided.

17 **MR. BABER:** Yes, your Honor. That's was where you
18 said, Well, you know, Oracle also was arguing that, well, maybe
19 the selection and arrangement of all these names taken together
20 could have some copyright protection. And you said, you know,
21 that's an issue for later, but for now each of the names, the
22 class names, method names, the package names are not protected.

23 **THE COURT:** Did I say -- well, all right.

24 **MR. JACOBS:** May I read, your Honor?

25 **THE COURT:** Yes.

1 **MR. JACOBS:** The right of completeness here, I think,
2 applies.

3 "In finding that the names of the various
4 items appearing in the disputed API package
5 specifications are not protected by
6 copyright, this order does not foreclose the
7 possibility that the selection or arrangement
8 of those names is subject to copyright
9 protection. See *Lamps Plus*."

10 The parenthetical on *Lamps Plus*:

11 "A combination of unprotectable elements --
12 italicizing unprotectable elements -- "is
13 eligible for copyright protection only if
14 those elements are numerous enough and their
15 selection and arrangement original enough
16 that the combination constitutes an original
17 work of authorship."

18 So we are -- we were back in originality land in
19 those days. I think we passed originality in this case by
20 stipulation. And there is no question about whether the
21 selection and arrangement of those names is an original work of
22 authorship. And now we're into, okay, infringement and
23 protectability under copyrightability doctrines.

24 **THE COURT:** Okay. We've got to bring it to a close
25 here. It's now 3:20.

1 Is there anything on any of your other motions
2 that -- going either way, that either side has got to have an
3 oral argument on? If so, we will do it, but I -- some of this
4 can be submitted on the papers.

5 **MR. BABER:** Just one, your Honor, that I'll mention
6 very briefly, but I think we discussed it many times, so it
7 doesn't need to be reargued.

8 On rangeCheck, where the jury found infringement
9 based on rangeCheck and you instructed the jury that for
10 purposes of that claim, the work as a whole was just the
11 arrays.java file in which rangeCheck appeared.

12 We believe that the proper test for infringement
13 always has to be the work as a whole as its registered. And if
14 so, then the nine lines of rangeCheck code is, as a matter of
15 law, diminimus.

16 **THE COURT:** Well, if you were right about that, yes,
17 but I don't think you're right about -- your position is that
18 the work is registered.

19 Now, that would be a -- I've read the cases. The
20 cases specifically reject that proposition and say that I am
21 supposed to identify what the work as a whole is, and it can
22 vary from work to work. So there's policy reasons that might
23 support your argument, but I don't think that's the law in the
24 Ninth Circuit, so.

25 All right. Is there any other one that anyone wants

1 to really argue?

2 **MR. JACOBS:** Your Honor, could I substitute just a
3 brief follow-up on an earlier discussion in response to the --
4 in response?

5 **THE COURT:** Okay.

6 **MR. JACOBS:** There is one issue that I wanted to
7 return to, just because I think the record wasn't accurately
8 reported to you, and that's on the question -- back on the
9 interesting question of fair use for some of the classes that
10 were part of the language specification.

11 **THE COURT:** All right.

12 **MR. JACOBS:** So there is -- there's a bit of outlier
13 testimony from Josh Bloch. He did the downstream packages or
14 downstream classes that you just elicited from Mr. Baber, that
15 that was in 10 packages. But every other witness said it was
16 60 or 61 classes, including Dr. Astrachan, who specifically and
17 several times agreed during cross-examination with Dr.
18 Mitchell's examination.

19 So we're talking again about fragments in terms of
20 the overall issue here.

21 **MR. VAN NEST:** Your Honor, I wouldn't want to put too
22 much reliance on that. Remember, we went to the jury on 37
23 packages as a whole. That's what we all agreed to do and
24 that's how it went in.

25 And our evidence on fair use is certainly by no means

1 limited to the point you raised. The point you raise is a good
2 one, and it may mean that for some of those packages the fair
3 use argument is even better; but it doesn't mean that there is
4 no fair use argument for the rest of them.

5 Our whole point is that when you look at all of the
6 factors taken together, we tried the case with the 37 packages
7 as a whole and Oracle at the end of the day withdrew any
8 request that they go package-by-package, and that's how they
9 went to the jury.

10 So I think the fair use case needs to be evaluated
11 on, you know, the merits of all the evidence on all the
12 factors, which go far, far beyond just the fact that some
13 number of these, whatever that number is, are absolutely
14 required just to use the language.

15 So, again, your point is a good one. It's correct.
16 But in terms of a JMOL, what we're looking at is the verdict
17 that the jury rendered and the question the jury answered, or
18 didn't, which affects all the packages taken as a whole. It's
19 the SSO of the 37 API packages, not just a few.

20 And that's -- that's the main point that I want to
21 make on that issue on JMOL.

22 **MR. BABER:** I have one tiny clarification and a
23 question. I promise.

24 **THE COURT:** What is this a tag team?

25 (Laughter.)

1 **MR. BABER:** No, no.

2 First, just to clarify what I just said about the
3 rangeCheck, that issue. We also believe that even accepting
4 your Honor's decision that the file, the individual file is the
5 appropriate work as a whole for rangeCheck, it's nine lines out
6 of 3,000.

7 **THE COURT:** But it gets booted up 20,000 times a
8 second.

9 **MR. BABER:** Understand, your Honor, but that's our
10 second level on rangeCheck.

11 The second is I just don't know whether your Honor
12 wanted -- you told us this morning you were curious about this
13 issue of in the source code, how I read --

14 **THE COURT:** Yes.

15 **MR. BABER:** Okay. The answer to that is, your Honor,
16 you have exhibits in evidence and testimony from Dr. Astrachan.
17 The answer is, no. They are not in the same order.

18 And if I can hand up --

19 **THE COURT:** Is there -- what have you got there?

20 **MR. BABER:** Well, what we have in the record, your
21 Honor, is we have several things.

22 First, we have from Java 5.0, we have Trial Exhibit
23 623.101 and what it is is a printout of all the source code in
24 the Math class, java.lang.Math.

25 **THE COURT:** I think I've got that right here.

1 **MR. BABER:** And we also have Trial Exhibit 47.101,
2 which is the same thing from Android, java.lang.Math.

3 **THE COURT:** This is just the Math one.

4 **MR. BABER:** It's just Math, your Honor, just one
5 package.

6 **THE COURT:** How many differences are there going to
7 be when I look at this?

8 **MR. BABER:** You're going to find a lot of
9 differences.

10 **THE COURT:** How about in the sequence?

11 **MR. BABER:** Well, I have a chart here, your Honor,
12 just to hand out. It's demonstrative. I will give one to
13 Mr. Jacobs as well.

14 (Whereupon, document was tendered
15 to the Court and counsel.)

16 **MR. BABER:** We just printed out the names as they
17 appear in order in the classes. And you'll see they are they
18 are very, very different. And we can use our favorite example
19 "max" and I can show you how that plays out.

20 **THE COURT:** So this is the sequence in which they
21 appear?

22 **MR. BABER:** Yes, sir. It's just -- it just takes in
23 order what's in the larger exhibit, just in order.

24 **THE COURT:** Okay. I -- all right. Go ahead and make
25 your point.

1 **MR. BABER:** Okay. So in Java, which is exhibit
2 623.101, Java.language.Math method appears on Page 15 of the
3 exhibit beginning at Line 782. Where it says what we have
4 there: Public static int max open paren int, et cetera.
5 That's the declaration of the method. Then there's the
6 documentation. Then there is the implementing code.

7 If we go to the Android file, 47.101, we find the max
8 method declared on Page 11 beginning at Line 555 of the code.
9 You'll see exactly the same thing: Public space static space,
10 et cetera. They are just in different places within the file,
11 although they, obviously, are the same, implementations of the
12 same method.

13 **THE COURT:** Where you have the arguments, are these
14 faithful to the way it appears in the code?

15 **MR. BABER:** Yes, your Honor. I think if you just
16 line it up with -- you have the code right there with you. For
17 example, if you look at Exhibit 623.101, you will see --

18 **THE COURT:** Where can I find double ABS in yours?

19 **MR. BABER:** I'm sorry. Where can you find what, your
20 Honor?

21 **THE COURT:** Where can I find double ABS. You see the
22 first one under Java is double ABS. So where is double ABS?

23 **MR. BABER:** In Java it's going to be --

24 **THE COURT:** I'm sorry. In Android.

25 **MR. BABER:** First one in Android is double ABS.

1 Double ABS is in the middle of the second page, about
2 two-thirds of the way down in the Java version.

3 **THE COURT:** All right. So that -- that's an example
4 where you have a "D" and they've got an "A."

5 **MR. BABER:** Exactly.

6 **THE COURT:** All right. What is -- this is just one,
7 one class out of many.

8 How many classes there are all together? 600?

9 **MR. BABER:** 6,000 I believe -- no, I'm sorry.

10 **THE COURT:** Classes in the 37.

11 **MR. BABER:** It's thousands.

12 **THE COURT:** 6,000?

13 **MR. BABER:** I believe it is 6,000. Because that, I
14 believe, was the testimony that in order to replicate the SSO,
15 you would need --

16 **MR. VAN NEST:** It's 600 or 700 classes.

17 **MR. BABER:** 600 or 700 classes, 6,000 or 7,000
18 methods sounds about right.

19 **THE COURT:** All right. Well, you were the one
20 that -- I don't want you to do bad math here.

21 **MR. BABER:** I've done that before, your Honor. I
22 don't want to do it again.

23 **THE COURT:** It's less than one-tenth of one percent
24 thing.

25 **MR. BABER:** And I think what this shows, your

1 Honor -- and your question this morning shows that the
2 hierarchical structure that we see in documentation and things
3 like that, that's simply so humans can find things. All the
4 computer cares about, is it in the right file. Because then
5 the computer knows if it's in the java.lang.Math file, it can
6 be in any order whatsoever and it can find it.

7 **THE COURT:** Let me ask you a different question.
8 You use -- which one of these is the Java language?

9 **MR. BABER:** The Java Platform, your Honor?

10 **THE COURT:** The platform.

11 **MR. BABER:** 623.

12 **THE COURT:** All right. 623, all right.

13 Using the 623, give me an example of an interface as
14 opposed to a method. I would like to see what one looks like
15 in the flesh.

16 **MR. BABER:** I don't know that there are any in
17 java.lang --

18 **THE COURT:** It's okay if there aren't any in here.

19 **MR. VAN NEST:** I have my file cabinet here, your
20 Honor.

21 (Laughter.)

22 **MR. VAN NEST:** I know where that is in the file
23 cabinet.

24 **THE COURT:** All right, in the file cabinet.

25 All right. How about a field then? You know, the

1 classes have fields. They have got methods. And they have
2 interfaces, and there is another one I'm leaving out. I know
3 what a method is. I can recognize a method.

4 I would like to be able to recognize a field when
5 it's called out in the -- can you show me one of those?

6 **MR. BABER:** Let me back up a minute because I can
7 tell you how to recognize, I think, an interface when it's
8 there if you're looking at code --

9 **THE COURT:** Don't do that.

10 **MR. KWUN:** Your Honor, right in the beginning of both
11 of these, actually, they define "Pi" and they define "E."

12 Well, so if you look in 623.101 the first page is
13 basically documentation of the class. And then you see the
14 author information, and then on Line 81 you see public static
15 final double e.

16 **THE COURT:** Right.

17 **MR. KWUN:** That's defining a field, which in this
18 case is a constant, which is used for natural logarithms, of
19 course, 2.718 and so on.

20 And then on Line 88 you see a definition of another
21 constant, which is --

22 **THE COURT:** So those are regarded as fields?

23 **MR. KWUN:** Yes, your Honor.

24 **THE COURT:** It's a field of one, really. That's --
25 is that what that means?

1 **MR. KWUN:** It's a field that has the name "E," and --

2 **THE COURT:** The value --

3 **MR. KWUN:** I'm not sure which of these words, but
4 either "static" or "final," I believe, means this cannot be
5 changed. Once I declare it, you cannot change that field,
6 which makes sense for a constant.

7 **THE COURT:** Okay, I've got it. Okay.

8 Now, "E" I know what that is. Natural logarithm. So
9 if you wanted to have the field that had like an array that
10 had, say, five numbers in it, could that work here, too? Would
11 that be the place you would define it?

12 **MR. KWUN:** Frankly, your Honor, I don't know exactly
13 how you would define the array, but you could do it there.

14 **MR. HWANG:** Yes, your Honor.

15 **MR. KWUN:** And, your Honor, you may remember from
16 trial there was some testimony from Dr. Reinhold about fields
17 and how you could have, for example, a field for a -- he would
18 find something called -- I think for car. He said you could
19 have a field of whether or not it was painted and what color it
20 was. It might have been an example like that. You could have
21 a field like that for an object, which is an object would be
22 something you create out of a class.

23 **THE COURT:** All right. If you go further down, the
24 very last line 104. It says: "Return StrictMath.sign." What
25 is StrictMath?

1 **MR. KWUN:** StrictMath, your Honor, is another class.

2 So this is saying we're returning --

3 **THE COURT:** Where would we find StrictMath?

4 **MR. KWUN:** StrictMath is defined, I believe, in
5 another file. It's not defined in here. But StrictMath --
6 what this is saying is you're not actually returning
7 StrictMath. You're returning the sign of a --

8 **THE COURT:** Right. I got that part. But I want to
9 try to find StrictMath.

10 **MR. KWUN:** It's not in here.

11 **THE COURT:** It's somewhere else.

12 **MR. KWUN:** Yes, your Honor. So what this is
13 saying --

14 **THE COURT:** So where would I find it?

15 **MR. KWUN:** You would need to look in a separate
16 class, and since it doesn't have --

17 **THE COURT:** And that's called StrictMath?

18 **MR. KWUN:** Pardon me?

19 **THE COURT:** The class is called StrictMath?

20 **MR. KWUN:** Yes, your Honor. Generally when you see
21 things that start with capital letters, the convention is
22 that's a class.

23 **THE COURT:** And if it's lower -- what if it's lower
24 case?

25 **MR. KWUN:** So the StrictMath period sign, what that's

1 saying is that the sign method that is inside the StrictMath
2 class is being used here.

3 **THE COURT:** If you look in the StrictMath, what
4 language is that written in?

5 **MR. KWUN:** Well, we would have to look at it to see.
6 It could have been written in native code, but as a general
7 proposition there would be something that would be in Java.
8 When you went there, it might say look somewhere else yet
9 again, which could be in another language.

10 **THE COURT:** I think if you look at it, you'll find
11 it's in native language.

12 **MR. KWUN:** Some of these are in native code and
13 you'll see before the method the modifier "native."

14 **THE COURT:** All right. Now you've helped me
15 understand what a field would be. That's the "E" and the "pi."

16 So find an example of an interface that's defined
17 here that is not a method.

18 **MR. KWUN:** Your Honor, I don't think the Math class
19 defines any interfaces. I can give your Honor an example of
20 what an interface is.

21 **THE COURT:** All right. Give me a simple example.

22 **MR. KWUN:** So we had this discussion now many weeks
23 ago, but you can have the interface of compare to. And the
24 basic --

25 **THE COURT:** Say again? Compare what?

1 **MR. KWUN:** Compare to.

2 **THE COURT:** T-o?

3 **MR. KWUN:** Yes. And the basic idea is that you, in a
4 variety of different classes, are going to have sometimes the
5 desire to compare two members or two -- excuse me, two objects
6 created out of that class.

7 So there's something called an interface that says if
8 you are going to be a comparable class, a class where you can
9 compare two objects, what that means is that you must have
10 within your class a method called compare to. So the interface
11 is called comparable. And when you declare in your class that
12 you implement comparable, what that is is that is a promise
13 that inside your class you will have a method called compare
14 to.

15 **THE COURT:** Why would you ever do that as opposed to
16 just using a method?

17 **MR. KWUN:** Well, two things. When you have an
18 interface, you still must have a method that implements that.
19 And I -- I don't actually know what the reason is of why you
20 have the interfaces. I just know what they are.

21 **THE COURT:** Is there another good -- I have reviewed
22 the Math one.

23 Is there another good example like this, a
24 side-by-side comparison that I could look at that -- in a
25 different context that would -- I don't care what it is. Just

1 one that's about this thick (indicating) that I could look at
2 each version to get a better feel for what's being contested?

3 **MR. JACOBS:** Well, we'd like you to look at Java.nio,
4 your Honor, and we can make sure that you have that, those
5 exhibits.

6 **THE COURT:** You have it right here? I will take it
7 right now.

8 **MR. JACOBS:** No. But that would be --

9 **THE COURT:** You have it?

10 **MR. BABER:** No. I have a different one for you
11 though.

12 **THE COURT:** What is that?

13 **MR. BABER:** I've got arrays.java, which is the
14 class from which -- I'll just show you --

15 **THE COURT:** You gave me this one. You gave me Math.
16 I want Mr. Jacobs to give me one he wants me to read.

17 **MR. JACOBS:** We will get it to you right away, your
18 Honor.

19 **THE COURT:** Is it about this thick?

20 **MR. JACOBS:** I think it's might be thicker.

21 **THE COURT:** Don't give me a big thick one.

22 **MR. JACOBS:** I know. But I think we're being -- the
23 simplicity of max and Math is distorting the analysis.

24 **THE COURT:** All right. You give my the one you want,
25 but thicker it is, the less I can read.

1 **MR. JACOBS:** Understood, your Honor.

2 **THE COURT:** Let me just say this. I'm going to deny
3 the motion for JMOL on fair use. And I'm not suggesting how I
4 would come out on it if I were deciding this as the trier of
5 fact, but I think there was enough on the way the jury was
6 instructed that it could come out the way it did.

7 And I think if we have to have another trial on it,
8 probably the way to do it is to figure out which pieces are
9 protectable, which pieces are not, and then have an analysis on
10 fair use that is limited to the parts that are protected.

11 So I hate to even contemplate the idea of another
12 trial, but if we get there, that's the way it will have to be.

13 But I don't think it would be right to grant a Rule
14 50 on fair use in favor of Oracle.

15 On the one about declarations are not copyrightable,
16 I don't have to rule on that now. I think that's part of a
17 harder project on the whole SSO project that I am working very
18 hard on, but I don't have an answer for you.

19 On rangeCheck and whether it's diminimus, I'm not
20 going to set the jury's verdict aside. They said it was
21 infringing and I think the records can be construed to support
22 that.

23 There is a couple more of these that I'm prepared to
24 rule on. The jury said that the documentation -- there was not
25 infringement on the documentation. I think the record supports

1 that verdict, so no Rule 50 there.

2 I want to think about the eight decompiled files. No
3 ruling on that yet.

4 Equitable defenses, no ruling on that yet.

5 Improper registration and no ownership, I'm going to
6 think about that as well, but I'll just say -- no ruling on
7 that yet.

8 I think that's -- that's all the items that were on
9 your motions.

10 **MR. VAN NEST:** Thank you, your Honor.

11 **THE COURT:** We are -- we're getting pretty close --
12 do we have that -- do we have the jury instructions ready to
13 give to counsel?

14 **LAW CLERK:** Yes, we have a draft, yes.

15 **THE COURT:** Right now?

16 **LAW CLERK:** It doesn't have the read-back part in it.

17 **THE COURT:** Oh, oh. I want you two to think about
18 the read-back part.

19 Do you want me to give an instruction on the jury can
20 ask for read-backs? Generally judges don't like read-backs,
21 but the Ninth Circuit has a recently new pronouncement that --
22 I, of course, salute when the Ninth Circuit speaks. I don't
23 ask questions. I just do what they say.

24 But here is what they say, is that the ordinary rule
25 is that you get a -- if the jury asks for a read-back, you read

1 back every word of what the witness says. So if you had a
2 witness on the stand for two days and they wanted to hear about
3 part of it, you do the entire two days.

4 Now, does that make sense? Of course -- I won't say
5 that. I would say, does that make sense? You can -- a good
6 argument is, no. It would be too long and then it would defeat
7 the purpose and that the judge ought to have more discretion to
8 isolate the part that really is responsive to what the jury
9 wants. And we always have to remember, a lot of these get
10 handed down in the context of a criminal case and there are
11 special considerations there.

12 But the way it's always worked for many years, as far
13 as I can tell, is that the lawyers are pretty good about
14 agreeing on what should be read back; but the problem is if you
15 don't agree, then we get into the problem of having to read
16 back the entire thing.

17 I want you to think about whether we suggest -- not
18 suggest, but we say to the jury that if they would like a
19 read-back, they can have it, but it may take some time and so
20 forth.

21 I need your -- I'd like to have your guidance on
22 that. So think, be thinking. I have been thinking about it
23 and if you can agree on language, then, of course, I would put
24 that in.

25 But except for that, I think we have a set of jury

1 instructions ready to give you. And probably on Friday we
2 should have the charging conference so that you can -- you can
3 be ready to argue this on Monday.

4 **MR. VAN NEST:** We will give it some thought, your
5 Honor. Absolutely. Thank you.

6 **THE COURT:** So how much longer do we have with the
7 witness on the stand?

8 **MR. JACOBS:** About 20 minutes, your Honor.

9 **THE COURT:** All right. And then your cross. Then
10 will that other missing witness be here tomorrow so that we
11 can --

12 **MR. VAN NEST:** He will be. Mr. Bornstein is
13 available tomorrow. I think unless we need to do it, I would
14 just as soon finish up with Dr. Mitchell, but I need to confer
15 with counsel on that. And then put Bornstein on and then our
16 case.

17 **THE COURT:** What does your case look like?

18 **MR. VAN NEST:** Looks good.

19 (Laughter.)

20 **THE COURT:** How long is it going to be? How many
21 witnesses?

22 **MR. VAN NEST:** Well, we have probably got six or
23 seven witnesses and, but I still think what you told the jurors
24 and then I did, too, about finishing the evidence this week is
25 right. Assuming that we get through Dr. Mitchell and Mr.

1 Bornstein tomorrow, I think we'll get a significant part of our
2 case in as well.

3 We have Mr. McFadden we'll be calling, a couple of
4 Oracle folks. We have Dr. August on the '104 and Dr. Parr on
5 the '520. And, you know, nobody is long.

6 **THE COURT:** I don't know, that sounds like we might
7 not finish this week.

8 **MR. VAN NEST:** No, I think we will. I'm going to
9 make every effort to do that. I would love to be able to get
10 the evidence in this week and then do the charging conference
11 and argue it on Monday. That is a good plan. And I think
12 we'll try to trim our case down to accommodate it, too.

13 **THE COURT:** All right. There we go. So you're going
14 to give me the IO, is that it, Mr. Jacobs? The IO version of
15 these?

16 **MR. JACOBS:** Nio or something that's manageable, your
17 Honor.

18 **THE COURT:** Great. I look forward to it. Okay.

19 **MR. VAN NEST:** Thank you, your Honor.

20 **THE COURT:** See you.

21 (Whereupon at 3:44 p.m. further proceedings
22 in the above-entitled cause was adjourned
23 until Thursday, May 10, 2012 at 7:30 a.m.)

24 - - - -

25

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

E X H I B I T S

<u>TRIAL EXHIBITS</u>	<u>IDEN</u>	<u>VOL.</u>	<u>EVID</u>	<u>VOL.</u>
971			3173	19
5			3178	19
20			3179	19
955, 295			3213	19
292			3215	19
302			3216	19
301			3230	19
27			3249	19
258			3251	19

I N D E XPLAINTIFF'S WITNESSESPAGE VOL.**RUBIN, ANDY**

(PREVIOUSLY SWORN)	3178	19
Direct Examination Resumed by Mr. Jacobs	3178	19
Cross Examination by Ms. Anderson	3181	19
Redirect Examination by Mr. Jacobs	3198	19
Recross Examination by Ms. Anderson	3205	19
Further Redirect Examination by Mr. Jacobs	3207	19
Further Recross Examination by Ms. Anderson	3208	19

RUBIN, ANDY

Video Deposition Played	3210	19
-------------------------	------	----

MCFADDEN, ANDY

(SWORN)	3211	19
Direct Examination by Mr. Jacobs	3211	19
Cross Examination by Mr. Kamber	3258	19
Redirect Examination by Mr. Jacobs	3263	19

SUTPHIN, BRIAN

(SWORN)	3265	19
Direct Examination by Mr. Jacobs	3265	19
Cross Examination by Mr. Van Nest	3271	19
Redirect Examination by Mr. Jacobs	3279	19

MITCHELL, JOHN

(SWORN)	3280	19
Direct Examination by Mr. Jacobs	3282	19

- - -

CERTIFICATE OF REPORTERS

We, KATHERINE POWELL SULLIVAN and DEBRA L. PAS, Official Reporters for the United States Court, Northern District of California, hereby certify that the foregoing proceedings in C 10-3561 WHA, **Oracle America, Inc., vs. Google, Inc.**, were reported by us, certified shorthand reporters, and were thereafter transcribed under our direction into typewriting; that the foregoing is a full, complete and true record of said proceedings at the time of filing.

/s/ Katherine Powell Sullivan

Katherine Powell Sullivan, CSR #5812, RPR, CRR
U.S. Court Reporter

/s/ Debra L. Pas

Debra L. Pas, CSR #11916, RMR CRR

Wednesday, May 9, 2012