No. 18-956

IN THE

Supreme Court of the United States

GOOGLE LLC,

Petitioner,

v.

ORACLE AMERICA, INC.,

Respondent.

On Writ of Certiorari to the
U.S. Court of Appeals for the Federal Circuit

JOINT APPENDIX VOLUME 2
PAGES 342-725

Thomas C. Goldstein
GOLDSTEIN & RUSSELL, P.C.
7475 Wisconsin Ave.
Suite 850
Bethesda, MD 20814
(202) 362-0636
tg@goldsteinrussell.com
Counsel of Record for Petitioner

E. Joshua Rosenkranz
ORRICK, HERRINGTON &
SUTCLIFFE LLP
51 West 52nd Street
New York, NY 10019
(212) 506-5000
jrosenkranz@orrick.com
Counsel of Record for Respondent

PETITION FOR A WRIT OF CERTIORARI FILED JAN. 24, 2019
CERTIORARI GRANTED NOV. 15, 2019

TABLE OF CONTENTS

VOLUME 1

VOLUME 2

The following decisions have been omitted in printing the joint appendix because they appear in the appendix to the petition for certiorari, beginning on the following pages:

[216] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Tuesday, May 10, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[401] A. I do.

Q. And you know what an API is; right? What does that stand for?

A. API stands for applications programmer interface.

Q. Uh-huh.

And you see in the highlighted passage, the second paragraph with the lines there, that there's some language I want to ask you about. Do you see that?

A. I do.

Q. And it says here that the AdWords API and AdWords specifications—do you see those words?

A. I do.

Q. And it says those are the intellectual property and proprietary information of Google. Do you see that?

A. I do.

Q. And you do expect people with these contracts to honor them, do you not?

A. We do.

Q. And it says here that, "Your right to use, copy and retain your copy of the AdWords API and the AdWords API specifications is contingent on your full compliance with this agreement." Right?

A. I do.

Q. It does; right?

A. Uh -huh.

[402] Q. And you consider these to be the intellectual property of your company; right?

A. Well, they're one of a very large collection, yes.

Q. One of a very large collection.

This isn't actually the only agreement where you treat APIs as the intellectual property of your company; right?

A. There are many, many different kinds of APIs. This is the AdWords API.

Q. Uh-huh. And let me show you 5121, which is another contract involving APIs. Do you see that?

A. I do.

Q. You're familiar with this; right? One of your company's contracts.

A. This is not a contract.

Q. Well, it's a terms and conditions; right?

A. Again, may I explain what this is?

Q. Absolutely.

A. Okay. So we built a set of services that programmers who we don't normally deal with can use. And we, for business reasons, had restrictions on the things that they could do.

When you would build that service, by using that service you were bound by this document.

Q. Yeah. So I said it was a contract. You seem to call it something else. What is it? Is the contract a license? How would you describe it?

[403] A. It's essentially an automatic license.

Q. An automatic license. Whatever we call it, you expect people to follow it; right?

A. We do.

MR. BICKS: So I would move this into evidence, Your Honor.

MR. VAN NEST: No objection, Your Honor.

THE COURT: Received in evidence.

(Trial Exhibit 5121 received in evidence.)

(Document displayed.)

BY MR. BICKS

Q. 5121. This is the terms and conditions. It says, "Personal and legitimate uses only."

Do you see that?

A. Yes.

Q. And it says there that, "The Google Web APIs service is made available to you for your personal, non-commercial use only (at home or at work)." True?

A. Yes.

It would be helpful if I explained the difference between an API and a Web service.

Q. You can do that in a moment.

A. Okay.

Q. But bear with me.

What we're talking about here, just so we're clear, is [404] this is the terms and conditions for Google's Web API service; correct?

A. That is correct.

Q. All right. And what you say here is only use it at home; don't use it for commercial reasons. Right?

A. That's correct.

Q. And there's no question in this case that when you have Oracle's API packages in the—and it's billions of phones that have been activated, right, in the history of Android?

A. Yes.

Q. When you have in those billions of phones those—the design of those API packages, that's for commercial reasons; right?

A. Uhm, we don't have—I'm sorry, would you ask your question precisely again.

Q. You're doing this because you're a for-profit company. And you're highly profitable with Android; right? In fact, you said that in shareholder statements.

A. We freely license Android. And we make our money on Search and other applications on top of Android as well as the iPhone.

Q. Right. You make your money on advertising search, on top of Android; right?

A. And on other platforms, yes.

Q. Right. And that's profit-making activity; right?

[405] A. Yes.

Q. In fact, you've said—and these are your words; not mine—"hugely profitable"; right?

A. We are—yes. Yes.

Q. All right. And back to what I was asking you about here is, this is your API service contract. I want to go to the intellectual property—

A. May—

Q. —section. And I want to ask you about that.

THE COURT: What are you trying to say?

THE WITNESS: I don't know what the protocol here is, but this is about a service we provide, not an API. There is a difference.

THE COURT: You said he can explain that at some point. You don't have to, but if you're going to get around to it, let him explain.

BY MR. BICKS

Q. Well, but you've got contracts where you treat your APIs as proprietary, do you not?

A. You'd have to show me such a contract.

Q. Are you telling us here today that you do not treat your APIs at Google as proprietary to your company?

A. There are millions of APIs. So you have to ask on a per-API basis.

Q. Well, you tell me which ones, because I don't know all

* * *

[420] Q. Right. And so we're very, very clear, the software that you were giving away here for free, that software had in it those API packages; right?

A. Again, we—the implementations that we made, we licensed freely. That's—"giving away" is vernacular for we license it freely.

Q. That may be your vernacular. But I'm going to break it down easy. You give it away for free; right? People don't pay for it; right?

A. Again, I'm trying to be very precise.

Q. Right.

A. There is a license that allows you to use it without having to pay for it.

Q. All right. So when you don't pay for it, can we agree that's for free?

A. Subject to the terms of the license, yes.

Q. All right. And within what you give away for free, it's the declaring code and the structure, sequence and organization of the 37 packages that we're here in this courtroom talking about; right?

A. Yes.

Q. All right. And so what you're explaining here is, We give the software away, but we make money from that.

And you're explaining how you do it here; right?

A. Uh-huh, yes.

[421] Q. And you're explaining to the shareholders of your corporation the truth; correct?

A. I do, yes.

Q. And you say here that, "The evidence is that the people who use Android search twice as much as everything else"; right?

A. Yes.

Q. And so, clearly, there's more revenue associated with those searches. And you say that; right?

A. I do.

Q. And then you say, "One other thing, of course, is if they're using Android operating systems the revenue that we share and the searches are shared with the operator but not with anybody else." Right?

A. Yes.

Q. And you didn't share any of those revenues with Oracle, did you?

A. No.

Q. And you say here "it's more lucrative," do you not?

A. I do.

Q. And "Not only is there more searches, and there's more ads, but it's also more lucrative." Your words; true?

A. Yes.

Q. So on that basis alone, you say Android is what?

A. It's "hugely profitable."

[422] Q. And you use the word "hugely." What do you mean by "hugely"?

A. Well, I'm trying to promote our platform.

Q. You're trying to promote your platform. But you're doing it on an earnings call, which is a public forum that's governed by the regulations of the government and the securities rules and things of that nature; right?

A. Of course it is.

Q. Right. And it's got to be truthful; right?

A. Yes, of course.

Q. And so when you say the word "hugely," tell us what you mean by "hugely."

A. Well, in general, Google is a very profitable company. But it's important to state that these profits are coming out of our Google Search.

(Reporter interrupts.)

A. They are coming out of our Google Search, which is where our money is made, as I said earlier.

Q. What you say here is, the more people who use Android, they search twice as much as anything else; right?

A. I am.

Q. Yeah.

And I was asking you questions about what you knew about Sun.

Let me show you Trial Exhibit 22, please.

[423] 22 is a document that you've seen before; right? EMG, executive—

A. I believe I've seen this, yes.

MR. BICKS: Move 22 into evidence, please.

MR. VAN NEST: No objection, Your Honor.

THE COURT: Received in evidence.

(Trial Exhibit 22 received in evidence.)

BY MR. BICKS

Q. Remember I was asking you information about Sun, and you were a little hazy on the details; right?

A. Uh-huh.

Q. So let's go to 22, and go to page 3.

MR. BICKS: Put it on the screen, please.

(Document displayed.)

BY MR. BICKS

Q. "Android/Sun final approval." That's the title; right?

A. It is.

Q. And if we go to page 3, there was a question posed: "Who are they?" And the question, they were talking about Sun Microsystems.

Does that refresh your memory about what you knew at that time?

A. I don't actually remember being in this meeting, but I'll accept that this—it's possible I was in this meeting because this is—this is what would have been presented to the group

* * *

[439] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Tuesday, May 11, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[581] Q. Right.

Because at that time, you had commercial relationships with many of the major handset carriers; right?

A. Not for Java SE. That was freely available on computers.

Q. Sir, how many contracts and licenses did you have with people in the handset company, handset world?

A. For a tiny version of Java, we had contracts with all the major handset manufacturers.

Q. And which handset manufacturers?

A. Nokia, Ericcson, Sony, many. I don't recall the number.

Q. And how many phones, mobile phones, at this time was Java in, ballpark?

A. Well, none of them were running SE. None of them were running desktop Java.

Q. Right.

And did you—are you familiar actually with the terms of your licensing to tell us here under oath that you didn't have licenses out to the handset manufacturers for SE?

A. To the best of my knowledge, we didn't. We had licenses for Java ME which was the micro edition, the tiny version of Java.

Q. Are you familiar with the license agreement with Nokia?

A. I don't recall the terms of it, no.

Q. Are you familiar with the license agreement with Danger?

A. No.

[582] Q. Uh-huh.

Are you familiar with the license agreement with Savage?

A. No.

Q. Do you know that Savage had Java SE in it?

A. I don't recall.

Q. I want to go back to your blog because I'm going to ask you about things that you said that were not in the blog.

Do you remember when the Google announcement came out that you referred to parts of it as crap, in your words?

A. Sure. I would not be surprised. I was very frustrated.

Q. But you didn't put some of these things in the blog; Right?

A. I'm not going to put every possible thought I've had in every blog I read.

Q. Right.

But the blog I think you were telling us was supposed to be an official statement of the company; right?

A. It was.

Q. And so I want to ask you about certain things that you were saying to executives at your company but didn't get into the blog.

You called the announcement, parts of the announcements, crap. Do you remember that?

A. We had private conversations. I had conversations with attorneys. I had conversations with our CFO. I didn't put

* * *

[585] Q. Do you recall in—right the day this announcement came out that you said that you had no

clue what Google was up to and your sense was that they were playing fast and loose with your licensing terms. Does that sound familiar to you?

A. Yes.

Q. Because you said it; right?

2368.

THE COURT: All right. Are you moving it into evidence? What is it—

MR. BICKS: I want him to look at it, Your Honor. But I would like to move it into evidence.

THE COURT: Tell us what that document is, Mr. Schwartz.

THE WITNESS: That document is a communication between myself and John Fowler.

THE COURT: Any objection?

MR. VAN NEST: No objection, Your Honor.

THE COURT: Received in evidence. You may put it up on the screen.

(Trial Exhibit 2368 received in evidence)

BY MR. BICKS:

Q. So this comes out right on the same day as the blog; right? A couple days after, maybe?

A. I think a couple days before. I don't have the date in front of me.

[586] Q. The blog, I think, is November 5th. This is November 7th. Do you see this?

A. Right.

Q. And you say, "I have no clue what they're up to. My sense is they're playing fast and loose with licensing terms."

And who were you referring to when you say they were playing fast and loose?

A. Google.

Q. Uh-huh.

And you didn't put that in the blog, did you?

A. We didn't have any clarity on what their licensing terms were. This was internal speculation. I didn't speculate on my blog.

Q. Uh-huh.

Well, you said you had no clue what they're up to. If you had no clue what they were up to, how could you be making an official company statement about what they were up to if you didn't know?

A. We wanted to be a part of the momentum they were building around making sure there was innovation in the handset community. That's what we were going to be a part of.

We didn't have any complete details, but there was a tremendous amount of PR surrounding their announcement, and we wanted to make sure we were a part of that PR.

Q. You wanted to be kind of on the stage, but, in fact, you [587] weren't, but you wanted to be; right?

A. Oh, absolutely.

Q. But it is an accurate statement that you had no clue what they're up to, and your sense was they're

playing fast and loose with licensing terms. This is what you said to Mr. Fowler; right?

A. Yes. It was before we knew what they were actually up to.

Q. And I asked you the question about whether or not you said their phones were lousy and—you recall that. Do you remember that, sir?

A. Was there a question?

Q. 5316.

A. What was your question?

Q. I asked you if you remember saying that the Google Android phone was horrible product, lousy or lame. Do you remember that?

A. I do not. And can I have a moment to read this?

Q. Absolutely.

A. (Witness reviews document.)

Q. This is something you wrote; right? Have you had a chance to look at it, sir?

A. No. I'm not quite done.

(Witness reviews document).

So what was the question?

Q. My question was I had asked you when I put up the document [588] where you had referred to parts of the announcements as crap and you said you weren't saying it about the phone, and I said actually do you remember making negative comments about the Android phone; right? And you said you didn't remember.

Now you remember; right?

A. Yes.

Q. All right.

So let's move 5316 in.

THE COURT: Any objection?

MR. VAN NEST: No objection, Your Honor.

THE COURT: Received in evidence.

(Trial Exhibit 5316 received in evidence)

BY MR. BICKS:

Q. And if we can go to the first paragraph there, this is something you wrote on May 2nd, 2009; right?

A. Yes.

THE COURT: Not May 2nd.

THE WITNESS: February 2nd.

MR. BICKS: February 2nd. Thank you, Your Honor.

Q. February 2nd, 2009; right? So this was after your blog; right?

A. Right.

Q. And you say here, "And to this day, even with a horrible product, it's Apple's iPhone versus Google's Android, even though the latter is lame"—those were your words; right?

* * *

[619] Q. One of the things you mentioned is that you had founded a company called Danger.

A. Yes.

Q. When did you found the company called Danger?

A. It was in December of 1999, I believe.

Q. And how long did you work there?

A. About four years.

Q. What, generally, did Danger do? What was its line of business?

A. We built, I think, what could be called one of the first smartphones.

Q. Would you describe that to the jury, please.

A. Sure. It was called the T-Mobile Sidekick. And it was a phone that was very, very good at accessing the Internet as well as being a regular phone that you could make phone calls with.

Q. When you say that it was a smartphone, are you distinguishing it in any way from some other kind of phones available at the time?

A. Yeah. I mean, the prevalent phones in the market were what we called feature phones. They had small screens.

In the 1999 time frame, the carriers didn't have any data networks. So you couldn't go buy a data plan from, like, Verizon or AT&T.

[620] So it was, kind of, on the cutting edge of—you know, Danger was experimenting on the cutting edge of what you could do beyond what the feature phones were doing. They were just good phones.

Q. And when you described that Sidekick phone as one of the—the first smartphone, I believe you said, is

it the kind of smartphone that we're familiar with today, the modern Android and iPhones?

A. More or less. I mean, it did a lot of the same functionality. It allowed to you surf the Web, get the full Web on a phone. It had a larger screen. The screen could be in landscape or portrait mode. It did instant messaging. It did email and things like that.

What it lacked is some of the user refinements like touchscreens and things like that. It didn't have a touchscreen.

Q. Did you consider that phone to have been as successful as the modern devices we see today on Android?

A. The scale is much different. I think in that era we sold about 2 million of the T-Mobile Sidekicks, which isn't very much.

Q. Isn't very much as compared to what?

A. As compared to what a smartphone would do today, which would be in the hundreds of millions per phone.

Q. When did you leave Danger?

* * *

[674] the third-party developer. Because, in these days, a platform without any third-party developer is an island. It never gets better. But if you have a third-party developer, you can continuously download new apps and get delighted throughout years.

So I felt the application framework, that's the area where Google really shines, is creating these functions for third-party developers so they can express

their creativity in a way that would create a great user experience for consumers. And that part was a lot of work. And there was a lot of new thinking there as well.

Q. Thank you.

You mentioned in your earlier testimony that some code had come from Linux, the Linux kernel, and some code had come from another open source project for the WebKit.

Could you explain to the jury other sources for where code that's part of the Android platform came from?

A. We talked about—well, previously we talked about licensing in code. And we talked about PacketVideo was the people who submitted how to play a video. An MPEG video, is what their code did.

But other open source projects, I think, were easy targets. They were existing open source projects. You know, they were already released. Everybody knew about them. So we would import one of those. Linux is the biggest one, but we [361] imported other ones as well.

Q. Have you heard of Apache Harmony?

A. Yes, I have.

Q. Does Apache Harmony code have any role with respect to the development of the Android platform?

A. That was one of the external open source projects that we imported into Android. And it was an implementation of the Java class libraries.

Q. And based on your experience at Android, what was your understanding as to the kind of license that you obtained Apache Harmony code under?

MS. HURST: Objection. Foundation.

THE COURT: You have to lay the foundation.

Why don't we stop here. It's 1:00 o'clock. Time to end for the day. And we will see the jury tomorrow at 7:30—I'm sorry, 7:45. 7:45.

Please have a good evening. Don't talk about the case. No research. No reading about the case. See you. Be safe, please.

THE CLERK: All rise.

(Jury out at 12:59 p.m.)

THE COURT: Be seated.

Mr. Rubin, you can take off. We need to have you back here at 7:30 a.m.

THE WITNESS: Thank you.

[676] THE COURT: All right. Thank you.

Before we break, are there any issues? There's one issue. Do you want to argue with me over the add-on rule on cross-examination designations that if you have more than twice as many as the other side's direct examination designations, and that number adds up to more than 30, meaning your twice as many is more than 30, then you have to do it eight hours earlier than the normal schedule?

So that will give everybody an incentive to lower the number of designations.

MR. VAN NEST: I think it's an outstanding idea, Your Honor.

Unfortunately, it doesn't help us at this point because it only restricts us going forward. They've already unloaded on us. I noted this morning four or five—they disclosed a lot of exhibits for Schwartz and these others, and yet half a dozen of them weren't disclosed.

So I'll live with what I have because, otherwise, I'm the only one that's going to suffer under the new rules. They've already—they've already done the damage at this point.

So I would ask the Court not to—not to burden me—

THE COURT: What does the other side say?

MR. BICKS: Well, we don't have to agree to it, Your Honor. But you know, this comment that you're using documents that aren't identified, when witnesses go beyond that which is

* * *

[694] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Tuesday, May 12, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[843] BY MS. HURST:

Q. All right. Let's look here at the bottom. It's an email from Mr. Schwartz.

Trudy, can you keep the Eric Schmidt part in, too? Just right there. Thanks.

All right. An email from Mr. Schwartz to Mr. Schmidt—right?—May 10th, 2007. So that's just two days after the big Sun open source announcement; right?

A. Yes.

Q. And Mr. Schwartz says to Mr. Schmidt: (reading)

"By the way"—

"BTW," that means by the way; right?

A. I believe so, yes.

Q. (reading)

—"we would of course love to work together. Our intent isn't to deliver a phone. It's to help others do so."

Do you see that?

A. I see that there.

Q. And then Mr. Schmidt forwarded that to you and asked for your comments; right?

A. Yes.

Q. And then you wrote back to Mr. Schmidt, "They have been calling me as well"; right?

A. Yes.

[844] Q. You wrote: (reading)

"I don't see any way we can work together and not have it revert to arguments of control. I'm done with Sun. Tail between my legs. You were right."

That's what you wrote; right?

A. Yes.

Q. And then you wrote, "They won't be happy when we release our stuff"; right?

A. Yes.

Q. And you wrote, "We now have a huge alignment with industry, and they are just beginning"; right?

A. Yes.

Q. You viewed Sun as a competitor now at this time; isn't that true?

A. Yeah. When we couldn't agree to be partners and I was about to release a clean room implementation of a virtual machine and class libraries and the whole operating system on top of it, this was a space that Sun was already in. They were selling things to the mobile industry, and by not—by basically, like, my failure to partner with them, turned them—turned us into a competitive nature. We were both targeting the same industry with similar products.

Q. And you wrote: (reading)

"I'm not underestimating their ability when folks like DoCoMo tell us they want to dump Sun for us. I'm [845] assuming we have something valuable and good."

Right?

A. Yes. That's what I wrote.

THE COURT: What is DoCoMo?

THE WITNESS: One of the main Japanese carriers like Verizon. They're government sponsored.

BY MS. HURST:

Q. Now, when OpenJDK came out—take that down, Trudy, please—it had on it what's called a GPL license; right?

A. I'm sorry. Is that a question?

Q. Yeah.

A. You know, I'm not sure exactly what the license was when OpenJDK first came out. I don't remember.

Q. Well, you certainly decided that the license that Sun used on OpenJDK was not acceptable to you; isn't that right?

A. Yeah. I mean, I recall at the time I looked at it, understood it, but I have forgotten a lot. Unfortunately, it's been a number of years.

Q. All right. But you looked at it, you understood it, and you decided the OpenJDK license was unacceptable to you; true?

A. I believe so, yes.

Q. All right. And you thought that phones could not be built using GPL software; isn't that right?

A. I thought there would be—it would be difficult for third-party developers to write their apps for phones if the [846] phones were based on GPL.

THE COURT: We need to remind the jury what GPL means. Explain in one sentence what GPL means.

THE WITNESS: Yep. It's called the GNU public license GNU, and basically it's an open source license that—one sentence?

THE COURT: What does it mean, GPL?

THE WITNESS: It—it's a viral license where—

THE COURT: A general public license?

THE WITNESS: GNU public license. GNU.

THE COURT: G stands for general, P stands for public, and L stands for license; right?

THE WITNESS: No.

THE COURT: No?

THE WITNESS: General is—it's the foundation that created it is called GNU, and it's the—they call it GNU, GNU public license, GPL.

THE COURT: At least the PL part is public license?

THE WITNESS: Absolutely right, yeah.

THE COURT: So that's what we're talking about?

THE WITNESS: Yes.

THE COURT: Thank you.

THE WITNESS: Sorry.

BY MS. HURST:

Q. Now, you said the GPL is viral. What did you mean by

* * *

[886] THE COURT: Sustained.

BY MS. HURST:

Q. When you left Google, did Android have a license from Sun or Oracle?

A. For Android?

Q. Yes.

A. No.

Q. All right. Did you write back to Mr. Lindholm and say, "Don't worry. We don't need a license because

Jonathan Schwartz put up a blog post in November of 2007 saying 'Welcome to the community'"?

MS. ANDERSON: Objection. Argumentive.

THE COURT: Sustained. Sustained. If you have more questions like that, I'm going to sustain the objection.

MS. HURST: Understood, Your Honor.

Q. Mr. Lindholm said, "We need to negotiate a license for Java under the terms we need"; right?

A. I see that.

Q. He did not write, "We can use the open source license for Java"—

MS. ANDERSON: Objection. Argumentive.

BY MS. HURST:

Q. —true?

THE COURT: Sustained.

[BY MS. HURST:

Q. Now, Mr. Rubin, this was not the first time that you heard somebody tell you that you needed a license for Java, was it?

A. I actually don't think that's what's going on with this email. I don't think he's telling me I need a license for Java. He was asked to look for alternatives by the founders of the company.

Q. Mr. Rubin, this was not the first time that you were told that you needed to take a license for Java, was it?

A. I—I don't think this is—this counts as a time I was told I needed a license.

Q. Let's just read it again: (reading)

"We conclude that we need to negotiate a license for Java under the terms we need."

Did I read that correctly?

A. You did.

Q. Were you at a company called Danger before you went to Android?

A. Yes. That's one of the companies I cofounded.

Q. And that was the Sidekick/Hiptop that we talked about?

A. Yes.

Q. And you put Java to SE APIs in Hiptop; is that right?

A. Yes. We created our own implementation of the Java 2 SE APIs for Hiptop.

Q And then Mr. Syzek at Sun came to you and he said, "You

[888] need a license for that", didn't he?

A. Yeah. I think the—I think the request was if we wanted to call it Java—and I did at the time, at the previous company—and I wanted to basically brand it with the logo, that I would need a license for that.

Q. You didn't have the Java logo on the Hiptop, did you?

A. On the Hiptop itself?

Q. Right.

A. On the hardware?

Q. You didn't have the Java logo on Hiptop, did you?

A. I feel like I need to explain that we didn't make the hardware.

Q. Just yes or no, Mr. Rubin. Did you put a Java logo on the phone?

MS. ANDERSON: Objection. Interrupting the witness' testimony.

THE COURT: The witness should be allowed to answer the question. Say yes or no and then explain.

THE WITNESS: No, there was no Java brand on it. It was a—we didn't build the hardware. We just provided the software. So I didn't have the authority to put a Java logo on somebody else's hardware.

BY MS. HURST:

Q. All right. You had what you thought was an independent implementation at Danger; is that right?

* * *

[924] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Thursday, May 12, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[954] (Trial Exhibit 7326 received in evidence)

THE COURT: Next witness.

MR. KAMBER: Good morning, Your Honor. Google would like to call Mr. Joshua Bloch.

THE COURT: By depo or screen?

MR. KAMBER: Live. He is here.

THE COURT: Do you want me to read it while he's here?

MR. VAN NEST: You can read it now, Your Honor. He doesn't have to be here.

THE COURT: The lawyers have an agreed-on statement that is like a stipulation. It's not very long. It will take two minutes to read it. This will be evidence in the case, so I'm going to read it slowly, and maybe raise your hand if you miss any of this because this will be the only time you will hear this. It may be read again.

Are you ready over there? You don't have to copy it down word for word, but it's going to relate to what the next witness is going to say. And both sides agreed to this, so this is evidence in the case.

"The Java platform is a software application platform that is used to write and to run programs in the Java programming language. The Java programming language is free and available to use. The Java platform includes, among other things, the Java virtual machine and the Java API packages. 'API' stands for application programming interface."

[955] Counsel, I'd like to read that paragraph again. Any objection?

MR. VAN NEST: No, Your Honor.

MR. KAMBER: No, Your Honor.

MS. HURST: No, Your Honor.

THE COURT: All right. I'm going to do it faster this time, but here we go.

"The Java platform is a software application platform that is used to write and to run programs in the Java programming language. The Java programming language is free and available to use. The Java

platform includes, among other things, the Java virtual machine and the Java API packages. 'API' stands for application programming interface."

Now, I'm going to continue on with their agreed-on statement.

"What is at issue in this case are the Java API packages, which are sets of pre-written computer programs used to perform common computer functions without a programmer needing to write code from scratch. These pre-written computer programs assist developers in writing applications. These pre-written programs are organized into packages, classes, and methods.

"An API package is a collection of classes. Each class contains methods and other elements. The packages, classes, and methods are defined by declaring code.

"The declaring code is the line or lines of source code [956] that introduce, name, and specify the package class or method. The declaring code allows programmers to understand and make use of the pre-written programs in the API packages to write their own programs.

"The declaring code for the packages, classes, and methods reflects the structure, sequence, and organization, sometimes called SSO for short, for the Java API packages."

So let me read that sentence again.

"The declaring code for the packages, classes, and methods reflects the structure, sequence, and organization for the Java API packages. The SSO specifies the relationships between and among the elements of

the Java API packages and also organizes the classes, methods, and other elements within the package.

"Each individual method performs a specific function. The declaring code for a method is sometimes referred to as the method declaration or", quote, "'header,'" close quote, "or," quote, "'signature',' close quote.

"The declaring code for a method tells the programmer the information the method needs, the inputs, to perform the desired function. Each method also contains implementing code. The implementing code provides step-by-step instructions that tell the computers how to perform the functions specified by the declaring code.

"The declaring code and the SSO of the 37 Java API packages at issue are protected by copyrights owned by Oracle.

[957] The copyright protection does not extend to the idea of organizing functions into packages, classes, and methods, but the copyright protection does cover the SSO as expressed in the 37 Java API packages."

So before we go any further, if you want me to read that again, or parts of it, raise your hand.

Okay. Four hands went up.

I'm going to go back to the part that I—I'm not going to repeat what I did read twice, but I'll go back, and this time I'll read a little faster, but I will read again what I have already read, except for the part that I have already read twice.

"What is at issue in this case are the Java API packages which are sets of pre-written computer programs used to perform common computer functions without a programmer needing to write code from scratch. These pre-written computer programs assist developers in writing applications. These pre-written programs are organized into packages, classes, and methods.

"An API package is a collection of classes. Each class contains methods and other elements. The packages, classes, and methods are defined by declaring code. The declaring code is the line or lines of source code that introduce name and specify the package, class, or method.

"The declaring code allows programmers to understand and make use of the pre-written programs and the API packages to [958] write their own programs.

"The declaring code for the packages, classes, and methods reflects the structure, sequence, and organization for the Java API packages. The SSO specifies the relationship between and among the elements of the Java API packages and also organizes the classes, methods, and other elements in the package.

"Each individual method performs a specific function. The declaring code for a method is sometimes referred to as the method declaration, header, or signature. The declaring code for a method tells the programmer the information the method needs, the inputs, to perform the desired function.

"Each method also contains implementing code. The implementing code provides step-by-step instructions

that tell the computer how to perform the functions specified by the declaring code.

"The declaring code and SSO of the 37 API packages at issue are protected by copyrights owned by Oracle. The copyright protection does not extend to the idea of organizing functions into packages, classes, and methods, but the copyright protection does cover the SSO as expressed in the 37 Java API packages."

All right. Maybe you will hear that again at some future point, but I think that's the best I could do for now.

Please call your next witness.

MR. KAMBER: Thank you, Your Honor.

[959] Google calls Mr. Joshua Bloch. He was just outside the door a minute ago.

THE CLERK: Will the witness please approach the witness stand.

THE COURT: All right. Are you Mr. Bloch?

THE WITNESS: I am.

THE COURT: Okay. Welcome. Please come up here and raise your right hand and take an oath to tell the truth.

JOSHUA BLOCH, DEFENDANT WITNESS, SWORN

THE CLERK: Please state your name for the Court, and spell your last name for the record.

THE WITNESS: Joshua Bloch, B-L-O-C-H.

THE COURT: All right. Great. Welcome. Please have a seat.

And you should adjust the mic so it's about this close to your voice. That's right. Say your name.

THE WITNESS: Joshua Bloch.

THE COURT: That's very good.

Counsel, go ahead.

MR. KAMBER: Thank you.

Again, ladies and gentlemen, my name is Matthias Kamber. I'm one of the lawyers for Google. Mr. Van Nest introduced me earlier this week, but it's been until today where I get—

THE COURT: Spell your name for the jury.

MR. KAMBER: Sure. It's M-A-T-T-H-I-A-S, H is silent.

* * *

[1003] the years?

A. I have.

Q. Dr. Bloch, I'd like to show you an exhibit that has been marked by the parties as 624.

A. Got it.

Q. Now, this Exhibit 624, do you recognize it?

A. I do.

Q. It's one of your presentations that you gave about designing APIs?

A. Yes.

Q. That's got your name on it and Google's name on it?

A. It does.

MS. HURST: Move the admission of 624.

RIGHT4: No objection, Your Honor.

THE COURT: Received in evidence.

(Trial Exhibit 624 received in evidence)

BY MS. HURST:

Q. Now, this particular presentation, this is one you gave at Javopolis. Am I pronouncing that right?

A. We say Javopolis, but—

Q. Javopolis. In 2005; is that right?

A. Yeah. I believe that was in Antwerp, Belgium.

Q. And who comes to Javopolis?

A. Just programmers, engineers.

[1004] Q. Okay. Let's look at page 2 of the document.

A. Got it.

Q. And is that a slide that you wrote?

A. Yes, it is.

Q. In fact, you wrote all of the slides in Exhibit 624; is that correct?

A. Almost certainly, but give me a second.

(Witness examines document.) Yeah. Other than the—other than the graphics on the templates.

Q. All right. And so you wrote, "APIs can be among a company's greatest assets"; right?

A. I did.

Q. "Can also be among a company's greatest liabilities." And there you explained "Bad APIs result in an unending stream of support calls"; right?

A. I did. Yes.

Q. And I think on direct you said that some APIs are harder than others.

A. Harder to—

Q. Harder to write than others?

A. Yes, some are harder to write than others.

Q. And why are some APIs harder to write than others?

A. Because of the complexity of figuring out how best to express what it is that the programmer wants done. Some tasks are more complicated than others.

Q. All right.

[1005] And one of the other things you said on this slide is that "Public APIs are forever. One chance to get it right." That's what—and I think you said something like that on direct here today as well; right?

A. No. I did not say that on direct today.

Q. Okay. Well, this is—you presented this; true?

A. That is true.

Q. And you believed that; right?

A. I believed it at the time more than I do now. In the interim, Java has taken a few more chances at some of its APIs. You end up with a little bit of a mess when you have multiple APIs trying to do the same things, but you can try and do it.

Q. Let me show you, sir, Exhibit 877, as well. Do you recognize that exhibit?

A. I do.

Q. And is that a document that you authored in or about September 2008?

A. Not really. I mean, it's derivative. It comes from the—what do you call that?—the abstract of a talk that I gave at OOPSLA in 2004 on the same topic.

Q. You published this particular version of it—

A. I didn't publish—

Q. —Exhibit 877, on or about September 7, 2008; true?

A. Technically, no. InfoCue published it.

Q. Did you authorize the publication by InfoCue in or about

* * *

[1129] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Monday, May 16, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1264] Q. All right. And you don't consider the source code that implements the API to be part of the API; true?

A. I try to be very careful in speaking about the API method declarations. I think that when one understands API, that that's widely used and means different things, depending on the context, but for the purposes of what we're talking about here, we're just talking about the API method declarations.

There are other contexts in which somebody might use the implementing code to be part of the API, but I wanted to be careful here. I'm trying to be careful in saying just the declaring code from the API. It's possible that in other uses, it might include the implementing code and the specification.

Q. Is it true, sir, that you don't consider the source code that implements the API to be part of the API; you consider that the implementation that's not the API?

A. I think for the purposes of what we're talking about here, that seems like a reasonable way of looking at it, yes.

Q. Now, the purpose of an API is to understand and use the implementing software; true?

A. I—I concentrate on use rather than understand. We don't really need to understand the implementing code to be able to use it. We need to understand it at a high level, but we don't need to understand the thousands of steps that might be needed.

I think what we need to understand is how to map the [1265] inputs to the outputs so that using the API is absolutely an important part of it.

Q. Is it true, sir, that APIs are ways to help understand and use software?

A. Absolutely.

Q. And is it true that the purpose of the 37 Java APIs in Android is the same as it is in the Java platform?

A. Well, I spoke earlier about how these method declarations are being used in a new context, and in

that sense, that purpose is different because creating an application on the Android platform is a different context than creating an application on the laptop or desktop computer. So at that level, the API purpose is different because I'm creating a different kind of program.

At a lower level, as I mentioned earlier the API has the sale purpose. It connects my code with the implementing code. That purpose is the same.

Q. It's true that knowing how the API is structured in Java will help in writing an Android program where the API is the same; correct?

A. Yes. That's correct.

Q. And you, sir, have offered no opinion on whether Android as a whole transformed the Java SE platform as a whole; correct?

A. I think that's correct, yes.

[1266] Q. Now, you've been in here in court in the past week; true?

A. Yes.

Q. And you heard Mr. Rubin say that he used the Java SE APIs in Danger; right?

A. I did hear him say that.

Q. He used it in the Hiptop and T-Mobile Sidekick; true?

A. That's what I understand, yes.

Q. And you also heard him say that those were smartphones; right?

A. I heard him say that they were early smartphones.

Q. And you know from your work in this case, sir, that there was a company called Savage that also used Java SE in mobile phones; true?

A. That was represented to me as part of my work, yes.

Q. And you also know, sir, from your work in this case that Sun licensed Nokia to use Java SE in mobile phones; true?

A. I—I accept that as true, sure.

Q. Now, designing a good API is difficult, isn't it?

A. Yes. The design process of creating an API is—is difficult for sure.

Q. And designing an API is hard in the same way that being an artist or a concert violinist is hard; isn't that true?

A. I think that's a quote from one of my depositions where I said that being a football player or a concert violinist or a ballerina, all those are hard things to do, and being an

* * *

[1268] is that correct?

A. Yes. I think the programming language and the APIs are used together, but they're different. Yes. That's reasonable, yes.

Q. And Google could have written its own different Java API; is that true?

A. I think if you're asking a technical question, would it be possible to rewrite APIs using something that was a completely different package and class organization, that from a technical perspective that's true. But it wouldn't meet developer expectations using the Java programming language.

Q. The choice to use the 37 APIs was not a requirement of the Java programming language; correct?

A. That's correct.

And I'm not sure if I'm supposed to answer with the other thing that we talked about or not here.

THE COURT: All right. I think both sides will agree that we're going to eventually get to tell you a stipulation that pertains to a small number of the 37, which I believe they're going to say was necessary to use the programming language.

Am I correct on that?

MS. HURST: Yes, Your Honor.

THE COURT: Is that correct over there, Mr. Van Nest?

MR. VAN NEST: I'm not sure how small, but yes.

[1269] THE COURT: Well, say in the range of three.

MR. VAN NEST: Yes.

THE COURT: All right. So something like that. We'll get you the details later. But that is what the witness is referring to.

So with that qualification, do you want to ask that question again?

MS. HURST: Yes.

BY MS. HURST

Q. With that qualification, sir, the choice to use the 37 APIs was not a requirement of the Java programming language; correct?

A. That's correct. It wasn't a requirement in the language. It was required to meet developer expectations in using the language effectively. That was required.

Q. Well, when you say "developer expectations," what you mean are Google's business goals; isn't that right?

A. No. I was referring to a general idea of to use a language effectively, as I mentioned earlier, you need access to libraries. So that's independent of Google or a specific company.

Even for my students to work effectively in the classroom, they need access to the libraries to be able to use the language to create software.

Q. Isn't it true, sir, that if my question is, In the world [1270] where Google didn't want to leverage a large body of software developers, and didn't want to leverage a large existing body of code and other libraries, if that wasn't something they wanted to do, they could have refrained from using the APIs and done something different? Isn't that right?

A. I think I understand the question, which is, is there a universe in which Google engineers could have used a completely different set of APIs?

And I believe there is such a place.

Q. And that is—place is particularly the place where Google doesn't want to leverage a large body of

software developers and leverage a large existing body of code and other libraries; isn't that right?

A. I don't think I'm—I know why Google would or wouldn't do something. I think it's certainly possible to create API packages with different labels. And then you wouldn't be using the same Java developers. So that's true. But I don't know what—why Google might do something.

Q. It's true, sir, that the Java APIs are viewed by the community at large as reasonably good APIs, aren't they?

A. Yes, I think that's true.

Q. And you agree they are good APIs?

A. I do.

Q. And not all software is the same, is it?

A. If you mean not all software is good, I agree with that as [1271] well.

Q. Because there's a notion of quality in software; isn't that true?

A. There is a notion of quality in software, absolutely.

Q. And some lines of code are better than others?

A. Sometimes it's tricky to go on a line-by-line basis. But, in general, some software is better than others. And sometimes you might be able to look at a few lines of code and say, well, those are absolutely better than some other lines of code. So sure.

Q. And that's true of APIs as well. There's quality in APIs; right?

A. There is quality in APIs, yes.

Q. And a good API makes a programmer's task simpler than a bad API; true?

A. I think that's a reasonable characteristic—one of the characteristics of what a good API is—that it does make a programmer's task simpler, yes.

Q. In fact, it is your view that without the Java APIs that Google took, Java programmers would have found it cumbersome to program for the Android platform; isn't that right?

A. Let me equate that to what I said earlier, which is, in using these API declarations, Android met developer expectations. If they hadn't, it would have been cumbersome to use.

* * *

[1322] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Tuesday, May 17, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1354] open source and all of that and the values behind open source and so you got to hear that, and now we're going to hear the other side which is the value of protecting intellectual property.

These are cosmic issues that are in play, and I'm not trying to diminish their importance, but both sides are going to have their say, so let's—I'm going to allow Ms. Catz to answer that question.

So please answer the question.

THE WITNESS: Most of the software industry, including the business software association, which works very, very hard—it's our trade group that is made up of nearly all of our competitors, and they all agree that it's very important to have the intellectual property, the brilliance that comes out, protected so that companies like ours and others can license that software to customers, receive money for it so they can invest back into it to build new software. The entire software industry is based on this.

BY MS. HURST:

Q. All right. You mentioned a license, Ms. Catz. What's a license?

A. A license is actually a permission to use or copy the software. It's written in a written document that actually lays out in quite a lot of detail exactly what the copier is—what their rights are, what they're allowed to do. And that's [1355] actually what we do with our product. We license it. It's not like a car where you just pick the car up and you hand it to someone. They give you money.

Because our software is all about taking a copy of what we've created, a license is—really explains in writing exactly what the user is allowed to do.

Q. All right. Ms. Catz, let's turn to a different subject. In the course of your business at Oracle, have you learned to what extent, if any, Android has had an effect on Sun and Oracle?

A. Well, Android's had a very negative effect on Sun Oracle in a number of ways.

Q. Would you describe those, please.

A. Sure. First, it's—it's basically forked the Java whole principle, the whole Java community has been forked, meaning split into two because the whole idea of Java, as I mentioned yesterday, is write once, run on any platform. That's really the—the bottom line.

And that meant that for developers, they could write it once, brilliant idea, write it once, and it would run on all sorts of different systems, which made it much more functional, much—not functional. Much more useful to those developers because they could—they could write it once and it would run anywhere.

Android—once you write it in Android, you can't run it [1356] on anything but Android. So—and if you write it in Java, you can't write it in Android. Now, that's one side of it.

Secondly, many of those customers that we used to license to to take a copy, to take a licensed copy like Samsung or ZTE or Motorola, just different licensees, BlackBerry, etc., they don't need a license—they don't take a license from us anymore because they use Android, which is free, and they end up using Android instead of actually paying us for a copy of—of our software.

Q. All right. Ms. Catz, let me have you look at Exhibit 5961 that's before you there. And do you recognize Exhibit 5961?

A. Yes. This is—this is one of the embedded business unit's budget—budget presentation that was given to us.

Q. And how are those budget presentations used within Oracle?

A. They're used as our way of planning for next year. They're also used to educate us on what is going on in the business.

Q. And are those budgets approved by the board of Oracle?

A. Yes. They actually go directly to—after they're approved by the Executive Management Team, they're actually presented directly to the Board of Directors.

MS. HURST: Your Honor, I offer Exhibit 5961.

MS. ANDERSON: Objection. Hearsay, Your Honor.

THE COURT: May I see the document? The objection is sustained so far.

[1357] BY MS. HURST:

Q. Ms. Catz, is the budget review a regularly-conducted activity at Oracle?

A. Yes, it is.

Q. And do you do it every year?

A. Yes, we do.

Q. Is Exhibit 5961 part of your budget review for the fiscal year 2011?

A. Yes.

Q. Was that presented to the management group of Oracle in preparation for adoption by the Board of Directors of the annual budget for fiscal year 2011?

A. Yes, it was.

MS. ANDERSON: Objection. Leading and—

THE COURT: Sustained. All of these are leading.

MS. HURST: Your Honor, it's foundational for the admission of a document.

THE COURT: When it's important, you are not supposed to lead.

There are too many slide shows in that document. If it was just a financial statement, I would allow it, but there are too many slide shows in that document to qualify it as a business record.

MS. HURST: Your Honor, may we offer one page from the document? I will point the Court to that page.

[1358] THE COURT: Yes. Please do that.

BY MS. HURST:

Q. Ms. Catz, would you turn to page 21 of the exhibit.

THE COURT: Objection still sustained. That's a speech. It's not a business record. That's a speech. So I'm sorry. You'll have to—

BY MS. HURST:

Q. Ms. Catz, in the budget review process for fiscal year 2011—first of all, was there anything significant about that particular year?

A. Yes. It was going to be the first full year that we actually owned Sun. We were—we were—we'd only owned them a few months previously. This would be the first full year going forward.

Q. All right. And were presentations made to you related to the Java business in that process?

A. Yes. It was made to me directly.

Q. All right. And what did you learn about the status, if anything—about the status of the Java business as part of the budget review in that first year?

A. What we learned is—and what I learned was that the Java business was being very heavily and negatively impacted by Android.

Q. And can you explain that, please.

A. Yes. The—the licensing business that this had always [1359] been going on was disappearing because a number of the—what were called OEMs or handset manufacturers were in fact adopting—had adopted Android and were not licensing Java anymore.

Q. And how, if at all, did that affect the profitability of the Java business?

A. It had a very negative impact. Companies like Samsung that would license a $40 million contract were down to—would be licensing a million dollars.

Q. How did that affect profitability?

A. Well, profitability—when revenues are here and expenses are over here, this period—this piece in between is your profitability. But when your revenues go down but your expenses are still high, still the same, your profitability goes away.

Q. All right. Did you ever have occasion to become aware of a situation with respect to licensing in Amazon?

A. Yes.

Q. What was the situation with respect to Amazon licensing?

A. So Amazon for the Kindle—it's a reader. For the Kindle Reader had used Java to create that Kindle Reader for many years. And then they had another product called the Kindle Fire and that one they used Android and so they didn't license Java at that time.

Q. And how did this situation come to your attention?

[1360] A. Because the way we look at different discounts and handle them with different customers comes through an approval process that came—comes to me, and so I was—I was made aware through that process that we—that—that—that basically Amazon was going to do the Kindle Fire with Android and that they were now considering a new product called the Paperwhite. It's called the Amazon Paperwhite, and they were considering whether to use Java for that or Android.

Q. And what happened?

A. Well, in order to compete with them, we ended up giving them like a 97 and a half percent discount for the Paperwhite. So instead of what we would have historically charged them, because our competition was free, they—they—we had to offer them a 97 plus basically couple cents on the dollar price from the list price.

Q. And overall, Ms. Catz, if you know, how have Oracle's Java licensing revenues in phones, mobile phones and similar devices, fared since Android?

A. They've done very, very poorly.

Q. Now, Ms. Catz, you do have the open source version of Java. Is that the reason for the decline in revenues?

MS. ANDERSON: Objection. Calls for expert opinion. This witness was not disclosed on that, Your Honor.

THE COURT: Well, those are two different points. Was she disclosed on this subject?

[1361] MS. HURST: No, Your Honor, but this is part of her ordinary business and understanding at the time, which was the—I believe the rule we heard last week.

THE COURT: Does she actually know the answer from her personal knowledge, or is this something that's been fed to her by somebody else?

MS. HURST: Personal knowledge, Your Honor.

THE COURT: All right. I'm going to let her answer then.

THE WITNESS: The phones that—that are in use by these other handset users use Android. They don't use OpenJDK. They are Android phones.

BY MS. HURST:

Q. Now, Ms. Catz, we looked on the timeline yesterday between the decision to acquire Sun and the closing of the deal. Did, at some point in that process, Oracle consider whether it might release a phone?

A. Well, we considered a number of things, but we decided that releasing a phone didn't make sense for us.

Q. Why not?

A. Well, as far as an actual phone, we were not in that business and that wouldn't have made sense for us.

Q. And what about more advanced phone software?

A. Well, as far as a phone platform, we looked at that very hard and realized that Android was just too far out ahead in [1362] the market and that it would be very difficult to compete with free, especially since they were using our software in it.

Q. All right. So what was your decision?

A. We decided not to do it.

Q. Is it true, Ms. Catz, that there are some Oracle products that work on Android?

A. Yes.

Q. And—

A. It is true, because Android has so taken over the handset market, if our applications didn't run on Android, our customers simply wouldn't be able to use some of our products, so we—we've ended up having to make them function on—on a number of devices, including Android devices.

Q. Ms. Catz, one of the considerations that the judge has instructed the jury is relevant to fair use is what would happen if everyone could do what Google is doing. What are your thoughts with respect to Oracle's business, if that were the case?

MS. ANDERSON: Objection, Your Honor, to the extent it's calling for a legal evaluation.

MS. HURST: Just as a business person, Ms. Catz.

THE WITNESS: Well—

THE COURT: Well, if the objection is legal evaluation, that objection is overruled.

Go ahead. Answer the question.

[1363] THE WITNESS: If everyone took a copy of our software without actually licensing it, we wouldn't have a business anymore. We simply wouldn't be able to afford to invest and—into the software industry if everyone else did what Google did, which is just take a copy of the software without a license. We wouldn't have a business.

BY MS. HURST:

Q. Ms. Catz, did there ever come a time completely outside the context of this lawsuit when you had an interaction with a Google executive about Google taking a license?

A. I'm sorry. Could you ask me that again?

Q. Sure. Was there ever a time when, completely outside the context of this lawsuit, in a social setting you had an interaction with a Google executive about Oracle's desire for Google to take a license?

A. Yes. Actually I was at a Bat Mitzvah and—which is like a Bar Mitzvah, but for girls—and Kent Walker, their general counsel, came up to me and said, "You know, Safra, Google's a really special company, and the old rules don't apply to us." And I immediately said, "Thou shalt not steal." It's an oldie but a goodie.

Q. When was this conversation?

A. This was in March of 2012.

MS. HURST: Pass the witness.

THE COURT: All right. Let's go to cross-examination.

[1364] MS. ANDERSON: Thank you, Your Honor. Just one minute. I'll set stuff up here.

CROSS-EXAMINATION

BY MS. ANDERSON:

Q. Good morning, Ms. Catz. My name is Christa Anderson. I'm counsel to Google.

A. Hello.

Q. I have a few questions for you.

You said you're the Co-Chief Executive Officer of Oracle; correct?

A. Yes.

Q. That's the parent of Oracle America; right?

A. Yes.

Q. And Oracle America is the new name for Sun today; right?

A. It is.

Q. All right. And as Chief Executive Officer, you're called upon to speak on behalf of your company from time to time; right?

A. Yes.

Q. You're a top executive of a very large corporation; true?

A. Yes.

Q. And you make those statements to employees of the company;

right?

A. Yes.

Q. And to the public in general; true?

* * *

[1396] specification does; is that true?

THE WITNESS: That—that is true. Although, I would say, you know, the central part of the specification really is—are those declaring statements. It is—it is—it is the thing that fundamentally defines the APIs that either applications use or independent implementers implement.

THE COURT: All right. So under this document that you're showing us now, Oracle, back at the time, had permission under this document to take the declarations and do its own implementing code; is that what you're telling us?

THE WITNESS: Well, okay. So there—there, of course, are important conditions in the license about creating an independent implementation. So the first paragraph is about the right to create applications that call the API. The second paragraph talks about the right to create an independent implementation.

MS. HURST: All right. Trudy, why don't we put the second paragraph up.

(Document displayed.)

MS. HURST: Would you like to continue, Your Honor? Or should I continue?

THE COURT: You go ahead. I may interrupt, but go ahead.

MS. HURST: Okay.

[1397] BY MS. HURST

Q. So I think the second paragraph, Mr. Screven, was what you were referring to in answer to the Court's questions. Why don't you explain those conditions—

MS. HURST: Trudy, can we get that over a little more to the left so we can read it clearly. Sorry.

BY MS. HURST

Q. All right. Why don't you walk us through those conditions you referred to.

A. Sure.

So this paragraph is Sun granting the right to create an independent implementation of the specification with the following conditions:

First of all, the independent implementation must fully implement the spec. And so that means that—that Oracle, as an independent implementer, would actually have to implement all of the declarations that are part of that API specification.

You know, that's—that is important because Java has this characteristic of write once, run anywhere, which means that if I build my Java application, I can run it in many different environments on many different implementations of the specification.

Number two, this double ii clause really further expands that idea, right, which is that your independent implementation [1398] must not modify, subset, superset or otherwise extend the license or namespace. And it goes on to talk about various kinds of Java programming language elements.

Again, that is a way to require implementations to fully implement the spec and not add anything to the spec.

So, in other words, if Sun's implementation of Java has a certain set of methods for the class file, then Oracle, when we do our independent implementation, we're not allowed to add things to the file, because an application programmer may become confused about what they can actually depend on between our two different implementations.

We also would not be allowed to actually remove a method from a file because, again, you know, that would mean that applications could no longer enjoy this write once, run anywhere characteristic.

Q. So let's just stop there for a minute, Mr. Screven.

There's a reference to "licensor name space." Do you see that?

A. Yes.

Q. What did you understand that to mean at the time?

A. Well, in the Java programming language, declarations are organized into a hierarchical namespace. So, in other words, you could kind of think of it as, like, a person's name is Edward Screven. So, you know, Edward is my short name, and Screven is my family name.

[1399] Well, in Java it's sort of similar. So there's a—more or less a family name called Java. And all of the declarations—you know, and there are many, many declarations as part of that Java name family.

There's also another name family called javax. There's another name family called com.Sun. So those top-level parts of that namespace are restricted under this license.

So I, as an independent implementer, am not allowed to actually put new names into that namespace, because programmers—application programmers assume that declarations that are within those namespaces are part of standard Java specification. And if they write their programs under that assumption, then when they try to run their application on someone else's Java platform, it won't work.

Q. All right. And then there's a third condition. Could you explain that.

A. Yes. The third condition says that the independent implementation must pass something called the TCK.

So TCK stands for Technology Compatibility Kit. The TCK is a collection of programs that actually call the specification API.

And if you—if those programs run correctly, the programs are designed to verify that the specification is—is correct, then there's a high level of assurance that the independent implementation actually does correctly implement [1400] the specification.

Q. All right. You've used this term "independent implementation" several times. Could you describe what that is, sir.

A. Yes. So as—as Your Honor was actually asking about, creating independent implementation means taking the Java programming language declarations that are part of the specification and then inserting—

you know, putting those into source files, Java source files, and then adding to those source files, you know, Java programming language statements that carry out the operations declared by the specification.

Q. Okay. Is that the same thing as a clean room?

A. No, no. That's totally different.

Q. Why not?

A. Well, a clean room is when I am intentionally trying to avoid having to satisfy some—some license terms or avoid having—trying to pay some money where I intentionally don't use a specification that requires a license.

So creating an independent implementation under this license is definitely not a clean-room operation.

Q. All right. You mentioned that it was important—these terms were important for preserving write once, run anywhere. Could you explain that a little bit more for the jury.

A. Yes. As a—as a programmer who's writing Java applications, one of the very most important aspects of Java is [1401] that if I write a Java application against the Java APIs, then I can run my Java application in many different environments. So not only in different operating systems on different types of computers, but also against different implementations of Java.

So if I build my application using Oracle's Java platform, I can also run it on IBM'S Java platform. I can also run it on Red Hat's Java platform. So as an application programmer that's very important because

it means I have the widest possible market for my application at what is really a low cost.

Q. And was that compatibility requirement—did you consider whether that was beneficial to customers because it promoted competition?

A. It definitely promotes competition. It promotes competition among Java platform vendors. And, also, it lowers the cost for application developers. So, you know, a small group of people can build an application which runs in many different environments on many different platforms.

I can tell you from Oracle's experience a long time ago, before we wrote software in Java we wrote software in C. So even though there are such things as C libraries, it was very expensive for us to actually run our applications on many different platforms. We had to do a lot of work to make them suitable to run on all those different platforms. With Java, [1402] we don't have to do that.

Q. And so what happened, then, if there's a lack of compatibility?

A. Well, what happens is basically I—I have to make one of two choices. Either I have to spend a lot of money to put my application on to different environments and different Java platforms, or I get locked into one single Java implementation.

Q. All right. You mentioned that TCK was a requirement of this?

A. Yes.

Q. What was a TCK?

A. The TCK is a collection of programs that—that—whose purpose is to actually exercise the API.

So, in other words, they call the—the method and other parts of the API specification in order to—and check the results from that—from those calls in order to verify that the implementation of the specification is—is correct.

Q. All right. So that's—that TCK is actually software also?

A. Yes. It's a collection of Java programs.

Q. All right. So is there a license with that software?

A. Yes. That software is licensed as well.

Q. All right. And would you describe the TCK license, please.

A. Well, it's—it's—there is a fee. It's a—it's a

* * *

[1410] the TCK. And we also—we also have commercial licenses to Java

Then there's also a separate offering, called OpenJDK, which is an implementation of Java, which is licensed under the GNU license called GPLv2 with Classpath Exception.

Q. All right. So you have both commercial licenses and open source licenses?

A. That's right. So we have a dual licensing strategy. So OpenJDK is available to encourage application programmers to learn and use Java.

And then we sell commercial licenses to companies that wish to have support for Java so if they have issues we can help them out, or if they wish to embed the license in devices and they don't wish to have to—to deal with restrictions imposed upon them by the GPL license.

Q. And what sort of restrictions are those?

A. Well, the GPL requires a party that uses software under the GPL to—to, first of all, make the source code available for—for—for that.

So, in other words, if I distribute OpenJDK to someone as a third party, I have to provide the source code to OpenJDK. But, moreover, if I change the OpenJDK to make it better in some way so that I'm adding value to it, I changed it to make it easier to embed on my smartphone, for example, then I have to give away those changes for free. Right? I have to [1411] actually make the source code of those changes available for free.

So, you know, we can sell commercial licenses to Java because many parties don't want to have to do that. They want to make their changes to—to Java and keep them for themselves at a competitive advantage.

Q. Well, wouldn't the Classpath Exception solve that problem?

MR. PURCELL: Objection. Foundation.

BY MS. HURST

Q. Are you familiar with the Classpath Exception as part of your OpenJDK license?

A. Yes.

Q. Would you explain that, please.

A. Okay. So the Classpath Exception says that just because I have written a program that combines the JDK with some other components, those other components do not have to be licensed under the GPL.

However, if I have changed the JDK, if I changed the implementation of the APIs, then those changes must be released under the GPL.

So the Classpath Exception does not permit someone to optimize the JDK and then charge for it.

Q. What, to your understanding, does it permit then?

A. It permits me to create applications, my own applications, and not have my components of the application be released under [1412] the GPL. It also means that it's easier for me to mix in components from other third parties.

Q. All right. So why do you have both of these types, both commercial and open source licensing?

A. Well, it's a business strategy. So we strike a balance between making Java as widely available as possible to developers, to people who want to build applications, and then charging for licenses for folks who want to embed Java. So ISVs and smartphone manufacturers and other kinds of device manufacturers.

So we have this thing called a dual-licensing strategy, which is actually pretty common in the open source industry, that lets us both widely promote Java, make it freely available to programmers who want to build applications or just experiment with it, and also charge companies that want to—to embed it.

MS. HURST: No further questions.

THE COURT: Maybe this would be a good point for our 15-minute break. Please remember the admonition.

(Jury out at 9:23 a.m.)

THE COURT: All right. Be seated.

Anything the lawyers need me for?

MR. VAN NEST: I don't believe so, Your Honor.

THE COURT: All right.

MS. HURST: Nothing.

* * *

[1427] Apache Harmony on the market?

A. IBM has Java licenses.

Q. IBM doesn't have a license to contribute IBM code back to Apache Harmony for Apache Harmony to freely distribute, does it?

A. Uhm, I don't think they have the right to—to—to pass on to third parties the—the rights that you're given under the specification license, TCK license, or commercial licenses.

Q. All right. And you also testified on direct about the terms of the specification license; right?

A. Yes.

MR. PURCELL: Could we get that up on the screen. That's Exhibit 610.1.

(Document displayed.)

BY MR. PURCELL

Q. This license was written at Sun by Sun; correct?

A. Well, I assume that Sun wrote it. I actually don't know if they wrote it or not.

Q. You weren't at Sun when this license was first published?

A. That's right.

Q. Now, I think you testified that the specification license is required to use the declaring lines of code that we're talking about in this case; right?

A. Well, it's the only license I know about that allows someone to create an independent implementation. It's—and [1428] it also grants application programmers the right to create applications against that specification.

Q. Mr. Screven, this specification license—isn't it your testimony that this specification license is the license that gives a developer permission to use the lines of declaring code?

A. Yes.

Q. All right. And declaring code is a form of source code. That's Oracle's position in this case; right?

A. Declaring code is a form of source code.

Q. All right. Now, the specification license requires that any independent implementation be compatible with Java; right?

A. Yes.

Q. And any independent implementation has to pass the TCK, the Technology Compatibility Kit; right?

A. Yes.

Q. And to be compatible and to pass the TCK, an implementation has to use the declaring code not just of the 37 API packages at issue here, but all the declaring code from all the API packages in the Java platform; right?

A. Yes. The specification license requires you to create a complete implementation of the specification.

Q. Right. So you have to use all of the declaring code, copy all of the declaring code word for word; right?

A. If you're creating an independent implementation, yes.

* * *

[1458] A. Oh, yes.

Q. Can you give us an example of how that might work?

A. So one example would be there is an API package in Java for logging—logging events. If a program is running for a long time, it can be useful to keep a record of what goes on in it, so an application developer will use an API to create the log messages that are stored somewhere for later analysis.

There's an API package in Java SE called java.util.logging. It is one of the packages in dispute. That's one way to do logging.

Around the same time, completely outside of Sun Microsystems, some other folks in the community decided they preferred a different approach, so they created an API package called Log4J. It solves

essentially the same problems, but the API is almost completely different.

Q. Let's talk a little bit about the designing process. If you're launching an API designing process, how do you begin?

A. Well, you—you—you begin usually in kind of a confused state, actually. You start out considering well, what are the—what are the broad problems that we want to solve with the API package, what are the technical limitations of the platform we're using, the hardware it's got to run on, things of that nature.

Once you've gone that far, what we often do is collect a set of use cases, which are terse descriptions of more precise [1459] kinds of problems that we would want to solve.

And then there is often, you know, a long period of brainstorming and writing on white boards and having meetings with colleagues and collaborators. Eventually, you get to a point where you can start writing fragments of declaring code that start to give shape to what will be the API.

I started writing fragments of declaring code for this package and the related packages, I don't know, a few months after—after starting work on it, but it changed immensely over time due to feedback. Once you get that initial declaring code, you get colleagues to review it, you get other people to review it, you get feedback, and then you start writing some of the implementing code so people can actually write sort of toy applications that attempt to solve some of the use cases and they can try to see well, how well does this work.

Q. Over the two years you spent working on designing the Java java.nio packages, how many drafts of the API were generated?

A. There were about 30 drafts.

Q. And you testified you've been working for 20 years in this field designing APIs. Do you consider the process of designing APIs to be creative?

A. Oh, it's intensively creative. That's why I like the *Harry Potter* analogy so much. It really is a lot like writing a book. You have to keep a lot of stuff in your head, and the end result is—is very, very rich and complex.

[1460] Q. Can you tell me about some of the creative choices that you make while you're designing an API?

A. Well, a lot of it is really about figuring out the structures that you want. Just going through the various elements, the kinds of API elements that Java lets you define. You know, first you have packages for the—for this java.nio.channels package and the four packages related to it, we had to decide well, how many packages should there be. We could have put everything into one package, but that would have been more difficult to learn, so we came up with a division of five.

Once we had the rough concepts in our head, we had to decide which classes we wanted, which interfaces we wanted. We had to decide how our classes and interfaces related, is one class a subclass of this one or is it a subclass of that one over there? Does a class implement some interface in this API? Maybe it implements an interface in some other API.

And then going down further, you know, there are all the methods. How do the methods relate? For every method, what kinds of inputs does it take? What kind of outputs does it produce? What kind of errors can it report? So methods can also be related to classes and interfaces in that way.

Q. How would you characterize the typical length of a line of declaring code?

A. Well, I think that's very hard to characterize. Some are [1461] very short and some are extremely long.

Q. Can you tell the jury about these examples of declaring code on this slide?

A. So these are four examples from some of the packages in the 37 in dispute. These are examples of declaring code of four methods. As you can see, there's quite a bit of text for each one. They say—they say quite a lot. And it took quite a bit of effort to design each one of these, along with all of the other methods and classes and interfaces that they relate to.

Q. So these are—these are single declarations on the slide?

A. Each—each one of these in the right-hand column—each line is the declaring code for a particular method.

Q. And what about names? Are they a creative part of the Java API design process?

A. Names are certainly a creative part. It's important to choose good names, but it's also important to use judgment. The thing about names is the—is the shorter the name, the nicer it looks, but there are

many fewer short names than long names just because with long names, you have more choices to make. So names are important, but they are not the be all/end all.

Q. Do you think it's accurate to describe the Java APIs as labels?

[1462] A. As labels? I think that's laughably simplistic.

Q. Why do you say that?

A. Well, a label is just a name. If all we had were names, then we wouldn't need any of the—of the structure and organization that you actually find in the API packages. Q. Is the Java programming language defined by a specification?

A. Yes, it is.

Q. And what—where would you find that specification?

A. It's actually a book called *The Java Language Specification*.

Q. And is that available online?

A. It is—most—most editions of it are available online.

Q. And do you have Trial Exhibit 984 up there?

A. No, I have something completely different here. 2237.

MS. SIMPSON: This is already in evidence, Your Honor.

Q. Dr. Reinhold, is that a copy of the Java Language Specification?

A. This is a photocopy—I'm sure it was quite tedious to make—of the book, of the Java Language Specification, Third Edition.

Q. And what version of the Java Language Specification does Java SE 5 require?

A. It requires this version, the third edition.

Q. Are the Java APIs part of the Java programming language?

* * *

[1470] since 1996?

A. Yes.

Q. And yet in all those years at Sun and Oracle, you'd never even heard of the term *structure, sequence and organization* or *SSO* until Oracle filed this lawsuit; isn't that right?

A. That's correct.

Q. The purpose of API design is to make APIs easy and intuitive for developers to use; wouldn't you agree with that?

A. I would say that's—that's a main goal.

Q. Now, if it wanted to, Sun could have made the Java SE SSO very simple; correct?

A. I don't think so, no.

Q. Well, let me give you a couple of examples. Sun could have chosen to put every single class into one API package, couldn't it?

A. Yes.

Q. All right. And Sun could have chosen to put hundreds or even thousands of methods in each class?

A. Yep.

Q. And Sun didn't want to do that, and the reason it didn't want to do that is because that sort of SSO would be very hard for developers to learn and use; right?

A. That's correct.

Q. And Sun didn't want its APIs to be hard for developers to learn and use?

[1471] A. On the contrary, Sun wanted them to be easy to learn and use.

Q. So Sun developed the SSO of the Java APIs based on what it believed would enable developers to learn and use those APIs most efficiently; right?

A. Yes.

Q. And in developing an easy-to-use, easy-to-learn SSO for the Java platform, Sun believed that that would enable it to draw more developers to the platform; right?

A. Yes.

Q. Now, could we get the list of the 37 API packages on the screen. Do we have the demonstrative? This is good enough.

So let me just ask the question, Dr. Reinhold. So there is a package, an API package in Java SE called java.lang; right?

A. Yes.

Q. And, in fact, most of the required lines of declaring code that you identified are contained in the java.lang package; right?

A. That's true.

Q. That's a package that contains information, methods, classes that are fundamental to the operation of the Java language; right?

A. It contains—it contains classes, interfaces, methods and so forth that are more closely related to the Java

* * *

[1474] A. Yes.

Q. And that package contains security functions; right?

A. It contains some of the security functions of the platform.

Q. All right. And Sun wanted to convey to developers by choosing that name that this was a place they could find some of the security functions in the platform; right?

A. Yes.

Q. Now, Dr. Reinhold, without APIs, the Java programming language wouldn't be much use, would it?

A. That's true.

Q. Without APIs, a Java programmer could write a program in the Java language, but that program couldn't then communicate with a computer monitor so you could even read the output; right?

A. Correct.

Q. And without APIs, a programmer could write in the Java programming language, but that program couldn't communicate with a printer so you could print the output; right?

A. Correct.

Q. Basically without the Java APIs, you could write a program in Java in isolation, but nothing you wrote could communicate with the outside world at all; right?

A. No. Without any APIs, you could not write a Java program that communicated with the outside world.

* * *

[1517] map?

A. No. I've used software visualization techniques throughout my career. Many of the books I've written have many diagrams that visualize the way in which classes and interfaces and other parts of software are organized and related to each other.

Q. So this is a standard tool?

A. Yes, this is a very standard technique.

Q. All right. So what do we see here on this first part of the map?

A. The first part of the map, kind of, gives the overall perspective showing all of the classes and interfaces in the Java API, which, as we heard—as we've learned, there are 166 of these packages. And these are all the classes and interfaces that are part of the Java API.

This is really just showing the organization, the way things are actually grouped together. We'll see the relationship in just a moment.

Q. So this is the whole set of classes and interfaces in Java SE 5 API?

A. Yes, that's correct. This is all the classes and—

Q. Approximately how many are there?

A. So there are an awful lot. I think there's in the order of thousands, 3,000 or so classes and interfaces. And there's 30,000-plus methods. Many, many methods. Many, many classes [1518] and interfaces.

Q. Okay. Why don't you continue.

A. So this diagram here shows an overlay of the 37 API packages at question in this case, and show roughly where they correspond to the various classes and interfaces that are part of the Java API.

So this really, kind of, gives a view of how the packages touch or are associated with the classes and interfaces that are part of the overall Java API.

Q. So is this all of the packages?

A. No. There's actually 166 total packages. And these are the 37 that are at issue in this case.

Q. All right. And what do you want to show us next?

A. So that's, kind of, a big-picture view. And sometimes when you see something from that 30,000-foot level, it's a little hard to get a feel for what's really going on inside.

So what I've done here is I've created animation that blows up or expands a particular part of the

software map. And this part of the software map shows the various classes and interfaces that are part of the so-called java.util package.

Let me just talk a bit about what a class is, and interface in this context. A class is essentially a small program that contains both declaring code and also implementing code.

And you'll see that the classes are shown here colored [1519] blue. They're kind of the blue circles or blue nodes or blue dots.

The interfaces are shown in green. Interfaces are basically lightweight classes that only contain declaring code. There's no implementing code associated with an interface. And that will become important when we get a little bit further along.

Q. Stop here for one moment.

Is this the methods?

A. No. This diagram does not show the methods.

Q. And why didn't you show the methods?

A. If we showed the methods on this diagram, it would be so much information being presented that it would be rather difficult to understand it. It would be just a lot of words on the screen. It would make it hard to see the design. The design essence and design organization and design relationships would be obscured to some extent.

Q. All right. What's next?

A. So what I would like to do now is show the relationships.

Up to this point, I've shown the classes and the interfaces, how they're organized together. But classes and interfaces are not islands. They don't exist in isolation. They're actually connected together in a very intricate web of relationships. And this diagram starts to show those relationships. In fact, this diagram shows all the [1520] relationships between the classes and interfaces that are part of the Java API packages.

Q. All right. So what are those gray lines?

A. So the gray lines are basically the relationships. And we'll talk a bit more. In a second I'll show you a zoomed-in view.

But at the high level, these gray lines show relationships between classes. So the classes relate to classes. Interfaces relate to interfaces. And classes relate to interfaces. So it's ways of showing the connection, the interconnected web of relationships between the various elements.

Q. All right.

A. What we're really showing here is the design. This is the declaring code. This is the part that's the design element.

Q. All right. Have you got more to zoom in on?

A. So now what I'm doing, at long last, is kind of getting into the heart or the core of what we're showing here in terms of details.

So I'm zooming in on the java.util package. And I'm showing a particular portion of the java.util package that relates to something called the Java collections

framework. And I'll use this as a running example throughout my discussion.

So the Java collections framework provides a set of classes and interfaces that allow programmers to store and [1521] retrieve data.

Imagine, for example, you have friends in your contact list. And you want to be able to find a friend's phone number. So you could use elements of the collections framework in order to do that.

So we're zooming in on just a portion of the Java collections framework and looking in more detail at some of the many classes and many interfaces and their interrelationships.

So what you're seeing there is, we have dozens of classes. Those are the blue parts. And there are gray lines connecting them, which means that they're also related to each other.

We also see dozens of interfaces, the green nodes. And there's lots of gray lines connecting those pieces.

And then there's lots of gray lines that connect the classes and interfaces together as well.

The key thing to note about this—and I'll show you some examples shortly—is that this diagram represents many of the creative design choices that the people who built the Java APIs had to consider when they came up with this particular way of doing things.

Could have looked totally different. They could have had different connections. They could have had different relationships. They could have had different classes, different interfaces.

They chose them to be this particular way. But I'll show

* * *

[1525] all these dependencies showing the package name, the class name. It would show the class. Let's say array, or ArrayList or Vector that we just looked at, and a relationship like extends to another class, like ArrayList or AbstractList. So we show the relationships.

Q. And did you look on the Android side as well?

A. So I took a report and analysis that was done by Mr. Zeidman.

Q. The one we heard about in the stipulation earlier?

A. Yes, yes.

So I took his report. He had, as you saw, 11,500'ish copied lines of code. And for every line of code that he identified in his report, I would then mark in this big output which of the relationships that were copied were ones that had come from Android's—Google's copying into Android.

So I ended up with a very large file with certain things marked as being copied.

Q. All right. Can you show us what kind of a map that resulted in?

A. Yes. So then I put that through my visualization tool, and that produced this diagram.

And what you see here is that all the relationships in the design that were copied are colored in red.

Q. All right. And what, if any, significance did you take from this mapping of what was copied?

[1526] A. So there's a couple of interesting things to note here.

First thing to note is the copied APIs, the copied classes, the copied interfaces, appear throughout the Java SE 5 API packages and classes.

So you can see, just looking visually, that they touch many, many different parts of this overall circle. The red is widespread throughout.

Q. And what—what significance did you draw from that?

A. That basically means what's copied is central or important. It shows it's—the copied APIs are all over from the API packages.

Q. All right. And did you—

THE COURT: Can I? Just, I want to mention something. These are just for illustrative; right?

MS. HURST: Correct, Your Honor.

THE COURT: All right. The jury doesn't know this yet.

When we have retained experts like this, they always show diagrams and things like this. But none of this will be in the jury room. They don't come into evidence. They are—the evidence is what the witness is saying. That is evidence.

But these illustrative diagrams and charts, and things that these experts use, are not in evidence. And they won't be in the jury room.

So I bring this up in case you see something that you [1527] think you want to remember, make a note about it. But you will not be—I wouldn't want you to fail to make a note because you think this is going to be in the jury room. Sometimes jurors get that point of confusion.

I really should have said something sooner.

So thank you. I'm sorry for the interruption.

MS. HURST: Thank you, Your Honor.

BY MS. HURST

Q. All right. You were saying that, I think—I don't want to paraphrase, but the red lines reflect the Android pieces, and it's throughout the platform?

A. That's correct.

Q. Did you draw any other significance from what you found when you colored in the lines for the Android piece?

A. Sure.

So if you look really carefully at what's colored red, you'll notice that some of the circles, the nodes, are red and some of the links are red.

The nodes that are colored red originally were either blue or green. So if there was a blue node that's now colored red, that meant that the class was copied.

And what that meant was that Google had copied the declaring code. And then, because it's a class, they provided their own implementation code.

On the other hand, if you see something that was colored

* * *

[1539] you did for Java SE 5?

A. Yes, ma'am.

Q. This is a subset of that?

A. It's a snippet. Yes, that's correct, subset.

MS. HURST: The text was inaccurate. That's why we removed it.

MR. KAMBER: As long as the witness doesn't testify about nonasserted—

MS. HURST: We're not going to do that.

THE COURT: All right. Thank you.

BY MS. HURST

Q. What does this show?

A. So this diagram illustrates the commonality between the Java security package—which is, as you can imagine, given the importance of security today, an important part of Java. It basically shows that the package design and purpose is the same between Android Gingerbread, in this case, which is one version of Android, and Java SE 5.

Q. Show us that.

A. Take a look here. You can see that the design is equivalent. So the same classes exist. The same interfaces exist. The classes being blue. Interfaces being green. The same relationships exist. So they're basically, fancy word, isomorphic. The two graphs have the same structure illustrating the same design.

[1540] Q. All right. If we think about that from a developer's perspective in the security package, what's the significance of it having the same purpose?

A. So it would mean that if somebody knew how to use security in Java SE 5, they would also know how to use this in the same way in Android. And, of course, we all know security is important. So that would be a valuable thing to have.

Q. All right. Have you done any further examinations related to this matter?

A. Yes. I've also taken a look at how—there were several full stack operating systems that were used in mobile devices that existed prior to Android.

And so the next couple of slides just walk through a couple of examples describing these full stack—

Q. Let's stop there.

On the left, what is this? Savaje?

A. On the left is a full stack operating system called Savaje. Kind of a funny name.

And Savaje was—I think that it was created around the 2002 time frame. And they produced a phone that was shown at the JavaOne Conference in 2006. That's where the "2006" comes from.

So it's a full stack operating system that uses Java SE. In fact, it uses all the Java SE packages that are at issue in this case.

[1541] Q. All right. And so tell us what this diagram shows.

A. So this diagram illustrates how the architecture, the layers of the Savaje full stack operating system

platform for mobile devices were essentially equivalent in terms of purpose to the layers in Android.

So as you can see there, both operating systems had Java applications. Both operating platforms had either full Java 2 class libraries or partial Java 2 class libraries.

Both of them had virtual machines. Both of them had native methods that are used to optimize the performance of the system.

Q. So that's the pink on the left and the green on the right?

A. Yes, ma'am. Native methods in both.

And they also both run on top of operating system kernels that shield and manage the different resources that are provided by all the hardware computing devices.

Q. So 2006, you're saying that's Java 2 SE on what kind of device?

A. So that would have been on a mobile device.

Q. All right.

And what significance, if any, do you draw from this comparison?

A. So what this says to me is that full stack operating systems were being used running Java SE prior to the release of the Android mobile device, mobile devices.

[1542] Q. All right. And do you have another example of that?

A. Yes, I do.

Q. Show us.

(Document displayed.)

THE WITNESS: So this is another example. This is an example of Java being used in a smartphone prior to the release of Android.

And what we see here is something called the T-Mobile Sidekick. This particular version that's shown up here, I believe, was released—it was the Motorola Q700 phone, I believe, released in around 2007.

MS. HURST: Your Honor, may I approach to hand the witness a couple of additional demonstratives?

THE COURT: Sure.

THE WITNESS: So this phone is the T-Mobile Sidekick. That's the one that's shown on the left.

MS. HURST: Hold on. We want to make sure we've got the right ones here.

THE WITNESS: Okay.

(Pause)

BY MS. HURST

Q. Did we give you the right ones?

A. I believe so, yes.

Q. Go ahead.

A. So this is the Sidekick. This is the one, you can see [1543] it's red. And this is the other one. This is the HTC Dream. This was the first phone that was released that ran Android in the 2008 time frame, as we see on the timeline.

And what's interesting about these two phones, aside from the fact they look very similar—they're very similar form factors. They both have sliders. They both have full keyboards. Both of them were using Java. And both of them ran apps. So you could use email. You could use Web browse. And you could use other kinds of apps, just the same way you could use when the HTC Dream came out.

So it basically shows that people were using Java on smartphones prior to the release of the Android hardware in 2008.

Q. All right. Were you here for Mr. Rubin's testimony?

A. Yes, ma'am.

Q. And did you hear him testify which version of Java was in Danger?

A. Yes.

Q. And what was that?

A. He said that Danger was running Java 2 SE, which is the same version which is at issue in this case.

Q. All right. Did you also have some opinions about creativity in the design of the Java APIs?

A. Yes.

Q. All right.

[1544] A. Absolutely.

Q. What do you have to tell us about that?

A. So if we think back to the big diagram that shows all the different classes and interfaces, and

their relationships, that diagram gave one view on creativity. It showed the intricate Web of relationships that connected all the classes and interfaces.

But if you look at that diagram, that kind of raises the question did those interfaces and classes and relationships have to look that way? Or was that the result of creative design that people had to think about very deeply?

And this example here illustrates one example—and I actually have several of them to show here—of how people had created equivalent capability, equivalent classes, equivalent functionality, and so on, that was available in that diagram we saw, but had completely different design structure.

And so the way to illustrate this is by using, once again, the Java collections framework that we were talking about before.

So the Java collections framework, if you recall, is the thing that allows you to be able to store and retrieve data that you might have in friends contact list or something.

And the java.util package in Java SE has the classes from the Java collection framework. All those classes exist in one package. There's many of these classes. There's dozens and

* * *

[1572] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 18, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1623] THE COURT: All right. Thanks.

BY MR. BICKS:

Q. This is a graphic that you're familiar with, Mr. Civjan?

A. Yes.

Q. And it talks about the ecosystem opportunities for mobile.

A. Yes.

Q. Can you explain to our jury—I think I asked you about the Java-labeled handsets. Who were those customers up at the top there, Nokia, Sharp?

A. Well, those are customers that licensed Java ME and Java SE for smartphones or cell phones. You know, all phones period. So Samsung, Motorola, LG. What we did on a chart like this was we would put some of the marquee names and we would try to cover the various geographies, but we, in reality, were in pretty much every manufacturer, every phone that was coming out, as you saw 85 percent, so people like RIM and SavaJe and Danger are not here, but this is an example of people who had licenses and were shipping our technology.

Q. So what I think you are saying is that this has a list of companies at the top, but does it exclude—include every company?

A. Absolutely, yes. It was an example.

Q. And so, for example, what are the companies that are not on there that Sun had licenses with?

A. Well, RIM, SavaJe, BlackBerry, as I said, Panasonic, Sony.

* * *

[1625] working with you?

A. I had a great team. It was—we had tenure, which in our industry is pretty short usually, but the tenure on my team was seven to eight years. The people were really smart, really technical, very dedicated to what they did. We loved what we were doing. It was a lot of fun to create something like this. It was a great team.

Q. Are you familiar with a company called Danger?

A. Yes, I'm.

Q. And did you execute a license agreement with Danger?

A. Yes, I did.

Q. Trial Exhibit 1026 should be in your folders there. Can you take a look at that?

A. Yes.

Q. Is that a license agreement you signed yourself, sir?

A. Yes, it is.

Q. All right.

I would move it into evidence.

MR. RAGLAND: No objection.

THE COURT: 1026 is in.

(Trial Exhibit 1026 received in evidence)

BY MR. BICKS:

Q. Mr. Civjan, is this a stand license?

A. Well, this is a standard license for compliance, so it licenses the JME technology to Danger to use in their cell [1626] phones, but we were really aggressive about finding people that were shipping our technology without a license. And—and what we would do in that case, like Danger, we found that they were shipping it and so we worked with them to have a compliance period where they could sort of have time to change the engineering and develop their product with the proper technology, with a license, pass the

TCK so it was compliant, and then ship and pay royalties, so this is that sort of license. It's a compliance-oriented lines.

Q. Explain to us the compliance issue that you were speaking of. Did that come up from time to time?

A. Yes, it did. I mean, people—Java you could download and use for research and development in the marketplace for free, but you couldn't ship it commercially without a commercial license, so people would sometimes do products, research and development, make a product, and then they would get a license, pass the TCK, which was really important because the *write once/run anywhere* was the promise of Java which you could write it on the Java platform and run it on any device that was running Java. So we would then have them become compliant, pass the TCK, and then pay royalties, as they should, for our technology.

Q. And how would you characterize Sun's and your team's attitude toward making sure that compliance happened?

A. Oh, we were aggressive. We were always looking for, you [1627] know, people that were noncompliant. Sometimes—most of the time I think it was by accident. They didn't realize they had to have a license; sometimes, you know, otherwise.

Q. And did you also execute a license with Nokia?

A. Yes, I did.

Q. And if you know, what Java product did Nokia license from Sun?

A. They licensed Java ME, Java SE for their phones.

Q. Let me go back, sir, to 9133.1, page 3. I want to speak with you for a moment about the products that Java was used in.

You see here this refers to mobile devices, TV devices, Blue-ray players, Java cards, and desktops. Can you share with us, sir, the products that Java was used in during this time period.

A. Well, we were—Java was used in pretty much everything that connected to the Internet at the time. So we were in smart meters. We were in PDAs. We were in cell phones. We were in set-top boxes. We were in TV's. We were on the desktop in computers. We were in E-readers. We were in VoIP phones with Cisco. We were in, geez, everything. Point of sale terminals. We were in printers. We were in the Playstation 3. We were in gaming machines. Everything that wanted to connect to the Internet was pretty much running Java

Q. And we picked out 5887.1, which was a pared-down version of an exhibit that I just wanted to have you look at.

* * *

[1633] A. Yes.

Q. And can you tell our jury the impact of Android on those licensing deals?

A. Well, Android—so basically Android was—or Google was talking to our customers. Our customers were switching to Android in various degrees, some very quickly, and they were switching developers

from Java to Android. They were switching phone designs from Java to Android.

And so the impact was, you know, less design wins, less future volumes, and less future revenue as they moved to Android instead of Java.

Q And what impact, if at all, was there on your licensing revenues because of Android?

A. Well, they were going down because they were displacing us in the phones or they were causing us to drop price to stay in the phones because they were free. They had a different business model.

Q. And explain to our jury the impact on price because of Android.

A. Well, we had to lower the price per unit for phones because in negotiations, they said well, we could ship, you know, Android for free, which basically is Java on Linux, and so why should we pay you, and, you know, we want it cheaper, and we'd have to drop price.

Q. And can you describe for us, if at all, how serious of a [1634] problem this was for you.

A. It was huge. It was devastating to our business.

Q. And did you continue to try to license the Java platform after the release of Android?

A. Yeah. That was our job as a group and a team, and we aggressively continued to license as much as we possibly could.

Q. And what impact, if any, did Android have on those continued licensing efforts?

A. Well, as I said, they displaced us in design wins so we got less phone design wins within companies as they used Android instead of Java, and also we got less revenue for whatever phones we still won or kept because there was price pressure and renegotiations of contracts as a result of Android being free.

Q. And if you look, for example, in Asian markets, from time to time, did you have to project losses because of the impact of Android?

MR. RAGLAND: Objection, Your Honor. Leading.

THE COURT: It is leading. You've done a very good job of not leading the witness, but here you are leading the witness. Sustained.

BY MR. BICKS:

Q. Did you at any time, if at all, perform any projections about the impact of Android revenues?

A. We did a lot. Basically I put together presentations on a [1635] regular basis to tell senior executives at Sun and then Oracle about the impact of Android, and I would collect the data from my team globally, from customers directly, and we would project the losses, so, yes, we did that regularly.

Q. And can you describe for the jury the projections that you saw and were involved in preparing. What did they show?

A. Well, there was, you know—we would show things like certain companies that were moving to Android, like HTC was an early adopter, Motorola, Sony Ericsson. We'd would show that they were going to have 50 or 60 percent of their phones be Android in the following year instead of Java.

We would show like an A-pack just China, Korea—I remember this was—it was China, Korea, and Taiwan each had a projection of $45 million of lost revenue over the three-year period. So we had data like this consistently, and it's in presentations I'm sure you guys have.

Q. You A-pack. What is A-pack?

A. That's Asia. With—A-pack for me—I ran Japan separately, so A-pack was Asia without Japan. So primarily the market for us in Asia was Korea, China, and Taiwan.

Q And on those A-pack markets, what, if any, impact did Android have on your business?

A. Well, the manufacturers were all moving to Android, and as I said, for that particular region, there was a $45 million projected loss for a three-year period and we were losing [1636] design wins and revenue.

Q. And you say *losing design wins*. Can you share with our jury what you mean by that?

A. Well, design win is—when somebody is making a new phone or a new anything, you know, all of the devices I mentioned, they decide what they're going to put on it, and they'll pick an operating system, they will pick a platform like Java to run on the operating system. And so that new design win is an engineering decision, and once it's made, that device, that phone, will ship with that decision.

And so when we lost the design win, instead of Java on that phone, they're going to use Android on that phone, and therefore we were displaced.

Q. Take a look, sir, if you would, on 4108, Exhibit 4108.1.

A. Point one?

Q. Yes. It's a spreadsheet. I think I have it on your screen because it's a PDF document.

MR. RAGLAND: Your Honor, we don't have 4108.1. We don't have that.

MR. BICKS: Here.

Q. Do you see that on the screen?

A. I don't have anything on the screen.

THE COURT: It's not on the screen.

THE WITNESS: I have a 4108 in a folder.

MR. BICKS: Let me get mine, the hard copy.

* * *

[1638] at, Mr. Witness?

THE WITNESS: I'm looking at what he put in front of me.

MR. BICKS: 4108.

THE COURT: Look at the one that has the label. That's the one that goes to the jury room.

THE WITNESS: Okay.

THE COURT: Tell us what the label says is the exhibit number.

THE WITNESS: Oh, this says 4108.

THE COURT: All right. So not 4108.1. Just 4108. All right.

And is that the one you intended to move into evidence?

MR. BICKS: Yes.

THE COURT: And what is the time period of that document?

THE WITNESS: This is—FY '07 actuals and then FY '08, '9 and '10 forecast, it looks like. And so this would have been—

THE COURT: Was this done back at the time—

THE WITNESS: This would have been done in 2000—

THE COURT: —in the ordinary course of business?

THE WITNESS: Oh, absolutely, yes.

THE COURT: By your group?

THE WITNESS: It was—our group was involved heavily [1639] in putting this together, but the document—this forecast was owned by product management.

THE COURT: Let me see.

THE WITNESS: It's a sales forecast basically.

MR. RAGLAND: Your Honor, the objection stands. It was done by a different department. There is no testimony that he was involved in preparing this document—

THE WITNESS: I was involved—

MR. RAGLAND: —when this document was actually compiled.

THE WITNESS: I was involved in this. I said I was.

THE COURT: Here we go. 4108 is received in evidence.

(Trial Exhibit 4108 received in evidence)

BY MR. BICKS:

Q. All right. Now, Mr. Civjan, you're a consultant today?

A. Yes.

Q. And we retained you to spend time to come—be prepared to testify?

A. Yes.

Q. And we haven't compensated you to be on this stand or anything like that?

A. No.

Q. All right. Can you tell us, sir, in one word the impact of Android on Java?

A. It was devastating.

[1640] Q. And how, if at all, did Android affect the morale at Sun and then Oracle?

A. It was a huge hit to morale. I was on the original Java team—

THE COURT: Wait. Wait. His answer was unclear. You said, *We haven't compensated you to be on the stand or anything like that*. So let's be clear. Have you been paid anything by anybody in this case?

THE WITNESS: I have been paid for my time in terms of opportunity costs like reviewing documents and things like that. For my consulting.

THE COURT: Who paid you?

THE WITNESS: Pardon me?

THE COURT: Who paid you?

THE WITNESS: Orrick, but not for testifying.

THE COURT: So you are drawing a distinction between before you came to court to testify versus your testimony today. You are not being paid for your time to testify?

THE WITNESS: Correct.

THE COURT: But before that, you were paid?

THE WITNESS: Yes, I was.

THE COURT: All right. So that's fair. Okay. It was the phrase *or anything like that* that threw me.

MR. BICKS: All right. Thank you.

THE COURT: I interrupted you in the middle of a [1641] question about devastating impact. Go back to that.

THE WITNESS: Could you repeat the question?

THE COURT: Mr. Bicks will ask it again.

BY MR. BICKS:

Q. Just tell us the impact of Android on the morale of your team.

A. So, yeah—so I was on the original team at Java in 1995, and we built it from the ground up, and I have a lot of pride in that, frankly. And, you know, Android was Java on Linux or is Java on Linux and was taking our customers, our revenue, and killing our business, and it was devastating to me personally and my team because we put our heart and soul into making this as successful—I mean, you guys have probably all heard

of Java and it's a huge success story in the industry, and it was hijacked. You know, they took our technology and they gave it away for free and they took our customers, and it was devastating.

Q. Thank you, sir.

THE COURT: All right. We will go to cross-examination.

MR. RAGLAND: Thank you, Your Honor.

THE WITNESS: Do I just leave this?

THE COURT: Leave those there for now, and they will collect them up when you're gone. Okay. Are you ready?

MR. RAGLAND: Yes, Your Honor.

[1642] CROSS-EXAMINATION

BY MR. RAGLAND:

Q. Good morning, members of the jury. I'm Steven Ragland, a lawyer for Google. I am playing myself today. I will try to get the lines right.

Good morning, Mr. Civjan.

A. Good morning. How are you?

Q. Civjan; is that correct?

A. Civjan, yes.

Q. Mr. Civjan, on direct examination, you referred to Oracle licensing Java SE to Nokia for a phone. Do you recall referring to that?

A. Uh-huh.

Q. You know, don't you, Mr. Civjan, that Nokia never actually put Java SE into a phone, don't you?

A. I am not sure honestly because I left, you know, the company, so I'm not sure what they have done since. Java—yeah. So I don't know if they did it, honestly.

Q. You know who Michael Ringhofer is; right?

A. He used to work for me in sales.

Q. He was the vice-president of global sales when you were at Sun; correct?

A. No.

Q. After you left, he became vice-president of global sales?

A. I don't know.

* * *

[1649] A. In the email, yes.

Q. Now, as—as Sun and then Oracle's vice president of worldwide sales, you knew that open sourcing Java through OpenJDK could reduce the company's competitive advantage from licensing the software, didn't you?

A. It didn't really impact the business.

Q. Well, you—you knew that it was Sun's strategy to build relationship with the open source community of developers to stimulate demand for commercial products that Sun sold; right?

A. Say that again.

Q. Sure.

You were aware that it was Sun's strategy to open source Java in order to build relationships with a community of developers and stimulate demand for Sun's

commercial products. You were aware of that strategy, weren't you?

A. Well, open source is a complicated issue.

Q. Mr. Civjan, I'll ask you to just answer my question and not give a speech.

A. I'm trying to.

THE COURT: Can you answer—

THE WITNESS: It's not a yes or no answer.

THE COURT: He says he can't answer it yes or no.

BY MR. RAGLAND

Q. You were part of discussions concerning how Sun's decision to open source Java would be received by companies like Nokia

* * *

[1684] Now, I'm not saying which one is correct. I'm sorry, who is that that's hacking and coughing? Would you like a cough drop, please?

MR. COOPER: It was me, Your Honor. I'm sorry.

THE COURT: Well, Mr. Cooper of all people.

(Laughter)

THE COURT: You get a cough drop.

Anyway, I'm not trying to tell you who is correct or not correct. That's for you to decide. But I'm going to let both sides have their say on this point.

All right. So it's limited to that use. 9198 received in evidence.

(Trial Exhibit 9198 received in evidence.)

THE COURT: Go ahead.

BY MS. SIMPSON

Q. Mr. Brenner, you signed this document?

A. Yes, I did.

Q. If we can look at the second paragraph of the exhibit, beginning with "Based on your presentation to us."

Do you see that sentence?

"It is apparent that the WIPI standard"—

A. Yes.

Q. Can you read that to the jury.

A. Yes.

Q. Slowly.

[1685] A. Yes.

"It is apparent that the WIPI standard," and then in parentheses, "(including the specification and any implementation of that specification) infringes Sun's intellectual property rights and violates Sun's Java technology licensing rules."

Q. Did you receive a response?

A. I did.

Q. What was your understanding of the position that this entity took with respect to the use of the Java software?

A. Their position, in response, was that they were not infringing on Sun's intellectual property rights because they obtained the technology that they based their work on from the GNU Classpath project.

Q. Did you agree with their position?

A. I did not.

Q. Why not?

A. Oh, I responded back to this organization that the GNU Classpath project was itself improperly—or unlicensed and also infringed on Sun's intellectual property rights; and, therefore, whatever they had done based on that work would also infringe on our property rights.

Q. Was this dispute ever resolved?

A. It was.

Q. In what way?

[1686] A. We engaged the U.S. trade representative, who in turn contacted the Korean government and persuaded them to stand down from this initiative of standardizing the technology developed by this research institution.

And they instead—that, instead, led to the licensing standard commercial licenses for Java by the Korean carriers and the Korean cell phone manufacturers.

MS. SIMPSON: Pass the witness.

THE COURT: All right. Thank you.

CROSS-EXAMINATION

BY MR. PAIGE

Q. Good morning, Mr. Brenner.

A. Good morning.

Q. Now, Mr. Brenner, you left Sun in December of 2006; correct?

A. That is correct.

Q. And that was almost a year before Google released Android in November of 2007, isn't it?

A. I believe that's correct.

Q. So you don't have any firsthand knowledge of what people said internally at Sun when Android was released; correct?

A. No, I don't.

Q. Now, you also left Sun years before Sun was bought by Oracle; right?

A. That is correct.

* * *

[1708] the requirements together, makes sure they're all met, and says *okay, it's okay to go ahead*; is that right?

A. Yeah.

THE COURT: You have to say it loud enough that we can get the court reporter to hear you.

THE WITNESS: I'm sorry. Yes.

MS. HURST: Thank you.

Q. And then after you worked on Apache JServ, you also worked on Apache JMeter; is that right?

A. Yes.

Q. And you worked on the Java Apache Mail Enterprise Server project?

A. Correct.

Q. And you were invited by Sun Microsystems to join the Servlet API expert group; is that right?

A. Yes.

Q. And after all your work on those various Apache products—projects, you were proposed for membership and accepted as a member of the Apache Software Foundation in 1999; is that true?

A. That's right.

Q. And you've been a member of the Apache Foundation ever since; is that true?

A. No. I left the foundation—I changed my status from membership to emeritus member in 2009.

[1709] Q. In 2009?

A. Correct.

Q. And have you, at times, also served on the Board of Directors of the Apache Foundation?

A. Yes, I have.

Q. Did there come a time, sir, when you became a mentor to the Apache Harmony project?

A. Yes.

Q. And a mentor was somebody who was promoting the project to try to get it started; is that right?

A. Yes.

Q. And later did you become a PMC of the Apache Harmony project?

A. No. The PMC is actually a group.

Q. Okay.

A. I was part of the—the Project Management Committee.

Q. So the Project Management Committee, that's what PMC stands for?

A. That's right.

Q. Were you a member of that Project Management Committee?

A. At the beginning of the project, yes.

Q. All right. And how many people were on that Project Management Committee?

A. I don't remember the exact number.

Q. Is it true that you have a Twitter account, sir?

[1710] A. I do.

Q. And have you ever described yourself as one of three people who created Apache Harmony?

A. I don't recall.

Q. All right. Let me approach you with Exhibit 9518. See if that refreshes your recollection.

Mr. Mazzocchi, does that refresh your recollection that you at one time believed that you were one of three people who tried to save Java by creating Apache Harmony?

A. That seems like I wrote that.

Q. And that was true at the time you wrote it; right?

A. Can't recall the state of mind I had when I wrote that.

Q. Is it customary—you wrote that on your Twitter feed; is that right, sir?

A. Yes.

Q. In or about 2010; is that right?

A. That's what this paper says.

Q. All right. And are you denying, as you sit here today under oath, that that was true at the time?

A. I'm sorry. Can you rephrase the question?

Q. Yes. Are you denying the truth of your statement in 2010 that you were one of three people who tried to save Java by creating Apache Harmony?

A. I don't deny it.

Q. Now, is it true that you've given presentations from time [1711] to time regarding the work of the Apache Software Foundation?

A. Yes.

Q. And did you give a presentation at ApacheCon in 2007?

A. I don't remember.

Q. All right. Is it true that the Apache Software Foundation is a registered nonprofit organization?

A. Yes.

Q. And that it's a volunteer organization?

A. Correct.

Q. And that all work is done by volunteers and nobody gets paid by the foundation?

A. There are a few people that get paid by the foundation to do mostly system administrator work.

Q. So that's like the IT crew for the foundation; is that right?

A. Yes. Exactly.

Q. Okay. And is it true that one of the purposes is to provide—of the foundation is to provide for open collaborative software development projects?

A. Yes.

Q. And is it also true, sir, that one of the purposes is to provide a means for individual volunteers to be sheltered from lawsuits directed at the foundation's projects?

A. I'm sorry. Can you repeat that?

Q. Yes. Is it true that one of the purposes of the Apache

[1715] A. Yes.

Q. I'm approaching you, sir, with Exhibit 5046. All right. Do you see there, sir, that this is an email thread with—in which you participated?

A. Yes. It appears to be the case.

Q. And it's an email thread in which you participated with Sam Ruby; is that right? If you look on the second line of the first page.

A. Right. Yes.

Q. And it's also copied to the mailing list members@apache.org; right?

A. Yes. It was sent to the members.

Q. Sent to them. Sorry. Exactly. And the email is dated April 17, 2008?

A. Yes.

Q. And you received this and responded to it at the time; is that true, sir?

A. I don't have any recollection of sending or receiving this email, but the document seems to indicate that's the case.

Q. Is your recollection sufficient for you to deny having participated in this email thread?

A. No. I don't deny it.

MS. HURST: Your Honor, I move the admission of Exhibit 5046.

THE COURT: Is this the one we talked about earlier?

[1716] MS. HURST: Yes, it is.

THE COURT: Has it been redacted?

MS. HURST: Yes, it has.

THE COURT: Received into evidence, 5046.

(Trial Exhibit 5046 received in evidence)

BY MS. HURST:

Q. I would like you to look at page 2—first let's look at page 3 of the email, Exhibit 5046, there at the very end. Do you see, sir, it's got your name. That's *Stefano*—I don't want to mispronounce it. I apologize.

A. Stefano.

Q. Stefano. Okay. Thank you.

And it has a single *greater than* sign next to your name. Do you see that?

A. Yes.

Q. And it's your understanding that that means—in the response where there's a single *greater than* symbol, that means it's something that you typed; right?

A. That's right.

Q. All right. So let's look at page 2, please. And in the middle of the page there, sir, you see there's a paragraph that starts with the words, "I was working" and there's a greater—just a single *greater than* symbol there. Do you see that?

A. I do.

Q. So correct me if I read this wrong, sir, but you wrote: [1717] "I was working under the assumption that we could ignore the trademarks, paren, avoid stating that we are compatible, end paren, use the org.apache.java plus classload trick to avoid the java.* namespace and pretend that we don't know of any IP that we infringe until explicitly mentioned. But what I was missing is the fact that the copyright on the API is real and hard to ignore."

You wrote that, sir; is that right?

A. I have no recollection of writing it, but this document seems to indicate I did.

Q. All right. And you further wrote: "Simply by implementing a class with the same signature of another in another namespace and simply by looking at available Javadocs could be considered copyright infringement, even if the implementation is clean room." You wrote that, sir; am I right?

A. Same answer as before.

Q. And you wrote, "So we are in fact infringing on the spec lead copyright if we distribute something that has not passed the TCK and, asterisk, we know that, asterisk." You don't remember writing that?

A. I don't. I write a lot of emails, and this was eight years ago.

Q. Well, sir, certainly being a member of Apache was something that you were proud of, wasn't it?

[1718] A. Yes.

Q. And you believed it did good work?

A. I did.

Q. And you wanted to satisfy the purpose of sheltering the individuals who participated from legal risks, didn't you?

A. Absolutely.

Q. And you, in order to fulfill that purpose, gave your honest opinion on the members@apache.org list; isn't that true, sir?

A. That is correct.

MR. KWUN: Your Honor, before we go further, I think we agreed there would be a limiting instruction associated with this document with the admission.

THE COURT: All right. I hope I get it correct. Correct me if I'm wrong.

This is a document that was written inside of the Harmony project, Apache project, about Harmony, and it is one of Oracle's responses to other evidence in the case in which Google has laid before you evidence that programmers thought it was okay to use the

declaring lines of code so long as they reimplemented, and this is internal to Harmony, at least so far, to show that arguably there was a counterview to that viewpoint.

And is that the limiting instruction?

MR. KWUN: Not for the truth of the matter asserted, [1719] Your Honor.

THE COURT: That's what I mean. It's not for the truth of the matter asserted. I should have started out with that.

This witness and whoever wrote this was not a lawyer. It's not being offered for the truth of what was asserted, and so you could not use this to conclude that it was copyright infringement or not. That would not be proper.

But it can be used to show the state of mind of somebody within the Apache project, which state of mind is relevant to meet the state of mind evidence as put in by the Google side on the same general subject.

So that's the limiting instruction. Thank you. All right. Let's continue.

MS. HURST: All right. Thank you, Your Honor.

Q. Now, sir, that email was in April of 2008. Was there a time in 2007 when you became aware of a dispute between Sun and Apache about whether Sun would grant a TCK license for the Harmony project?

A. Yes.

Q. And is it true that there was a debate within Apache as to whether Apache might go ahead and release the Harmony code anyway?

A. Yes.

Q. And is it true, sir, that you, in the context of that [1720] debate, said that you would never consider such an option?

A. I don't recall having stated that.

Q. Is it true that you were personally happy to just keep on—

MR. KWUN: Your Honor, I believe she is reading from a document that is not in evidence.

THE COURT: You can't do that. You have to ask him what he said. When he says he doesn't remember, show him the document, but just reading from that is too transparent so—

MS. HURST: All right, Your Honor. I'll do that.

THE COURT: You can use it to refresh his memory as to what he said at the time.

MS. HURST: Thank you, Your Honor.

THE COURT: But it has to actually refresh his memory.

BY MS. HURST:

Q. Mr. Mazzocchi, please go ahead and read the email and let me know when you're finished.

A. (Witness reads document.)

Okay.

Q. Mr. Mazzocchi, is it true that in or about April 2007, when there was a dispute between Harmony and Sun, you believed that it would not be the right thing to do to release Harmony and risk liability for both Apache and the users of the Harmony project?

MR. KWUN: Your Honor, this is still basically the [1721] same as reading the document.

THE COURT: Well, let's get the answer from the witness first and then I'll give an admonition to the jury. All right. What's the answer?

THE WITNESS: I'm sorry. Can you repeat the question for me?

BY MS. HURST:

Q. Absolutely. Is it true that in or about April 2007 when there was the dispute between Sun and Apache, you believed that it was not the right thing to do to release Harmony and that you would prefer to just keep working on it internally rather than risking both Apache and the users of Harmony liability?

A. The Harmony PMC was tasked to implement a fully compatible Java implementation, and we knew we had to pass the TCK to do that, to call it Java and to be compatible and have access to the entire set of IPs. We were focused on that at the time.

THE COURT: All right. Here is the admonition. Remember, I told you at the outset of the trial that what the lawyers say is never evidence. So Ms. Hurst asked a very specific question. The witness gave not so clear an answer, but—he didn't quite answer the question, but he did give an answer. It's the answer that the witness gave that counts and not the suggestion made by Ms. Hurst in her question. That's not evidence. Now—at least at this point.

[1722] BY MS. HURST:

Q. Mr. Mazzocchi, is it true that in the course of that dispute with Sun, it was your view that Harmony

should not be released, risking both Apache and the downstream users? Is that true, sir?

A. Yeah. We couldn't call something Java that wasn't Java.

Q. And later, as we saw in Exhibit 5046, you realized it was not just a trademark problem, it was a copyright problem, too, didn't you?

A. There was an entire set of intellectual property that Sun was using as leverage for the whole Java Community Process. That included trademarks, copyrights, and patents. That was my understanding at the time.

Q. All right. And then there came a time, sir, when you were aware that Oracle had acquired—had announced that it would acquire Sun; is that right?

A. Yes.

Q. And we have a stipulated timeline here on the board between the parties that that was in April 2009. Is that consistent with your recollection?

A. I honestly don't remember the timeline.

Q. Let me approach with 9201 to see if that helps you remember.

A. Thank you.

THE COURT: And the question is the time of the

* * *

[1761] But there were significant factors that were, sort of, threatening that window.

We've talked about the platform economics, the fact that coming late to the game is a disadvantage, and that if you're behind, you may never catch up.

We've talked about the fact that mobile use was rising, which meant that if Google didn't succeed in getting mobile users to use its search in the same way that desktop users had, it was going to lose advertising revenue.

There were rivals who were trying to do the same thing. And one of those rivals was the iPhone, which really, sort of, accelerated that process.

So this window of opportunity was potentially closing against Android.

Q. And can you actually identify the date the window opened and closed?

A. No.

You know, from an economic perspective the significance of the idea of a window of opportunity is precisely that you don't know exactly when it's going to close. You're sitting there. You know that at some point you might get locked out, to use that phrase. But you don't know when that's going to happen. You just know that the longer it takes, the more likely it is that you're going to fail.

Q. And to what—to what extent, if any, was Android [1762] successful?

A. Android was enormously successful.

Q. And have you prepared a graphic that outlines that?

A. Yes.

This is prepared from data that was collected in the context of this case, and shows that Android went from very low revenue in 2008, when it was introduced, to almost—a little over $18 billion in 2015. And the cumulative total over that period is over $42 billion in revenue.

Q. And where do these revenues—what are the different revenue sources there that you're identifying?

A. So as you can see, the revenue comes from ads, from apps, from hardware and digital content. By far the largest is ad revenue and apps.

Q. Uh-huh.

A. The others being smaller.

Q. So if Android—if Google doesn't charge for Android, then how do they generate so much revenue?

A. Well, basically, you can—you can see it on this picture. The revenue comes from the advertising. The presence—the fact that people have Android phones, the users of those Android phones go to the Web in—in ways that Google is able to monetize through advertising.

Q. And did you find any Google document where Google was looking at the strategic issues that you've been talking about?

[1763] A. Yes.

MR. BICKS: And maybe we can pull that up.

(Document displayed.)

THE WITNESS: So this is, kind of, a complicated graphic. This is an internal strategy document from Google.

And I'm not going to go through it in great detail, but I just wanted to illustrate the fact that, again, it shows Google thinking about, I think, in the exhibit, these issues using the same sort of phrases that I do.

They talk about the ecosystem building. They talk about the desire to extend the core business. New monetization services. And at the end they also talk about the importance of scale.

BY MR. BICKS

Q. And from an economist, why is this important to you?

A. Well, again, it's just confirmation that this model that I have of the economic forces that were operating on Android seems to be correct.

Q. And then as an economist, what is your opinion as to the—what extent, if any, Google's use of the Java API packages is of a commercial nature?

A. So I think it was highly commercial. It was very central to their overall business strategy as a company. And it produced a very large amount of revenue.

Q That was the 40-plus billion on the last graphic?

* * *

[1774] full stack as that term has been used here?

A. No. It's really not relevant.

As we talked about, the creator of the intellectual property licenses that property in whatever way they think is going to create value for them.

And what Sun and Oracle had historically done was to license Java to other people to make things. And

that was the business that was harmed by the emergence of Android.

Q. And did you make any attempt to measure or depict the harm to Oracle and smartphones?

A. Yes, I did.

(Document displayed.)

Q. All right. And can you explain to us what this shows?

A. So this is just a graphical presentation of data from a data service that does this kind of analysis for customers on shipments of phones. And this is showing the market share, the fraction of phones sold.

The blue line is the fraction of phones sold that had Java licenses. And the green line is the fraction of phones sold that were Android phones.

And what we see—and I should say, this is smartphones. So I'm not looking at the whole phone market here. But it's been suggested that, you know, Java was somehow not suited for smartphones, or that its decline had to do with the emergence of smartphones.

[1775] What I'm showing here is in smartphones specifically—so I'm just looking at smartphones here—what we find is that Java had, according to this source, something like almost an 80 percent share in 2007, which then, particularly when Android took off in 2010 and 2011, declined precipitously.

So that by the end of this period, Java's share in smartphones is—is nil. And Android has taken over that same approximate 80 percent share.

Q. Did you consider other markets besides the smartphone market to evaluate harm?

A. I did. In particular, eReaders and tablets.

Q. And has—how has Oracle experienced harm in the tablet and eReader markets?

A. Well, Ms. Catz talked about this with respect to the Kindle. Java was in the early Kindles. It's my understanding that Java was technically appropriate for the Kindle Fire, but that Amazon chose to use Android for that instead.

Kindle is still a Java licensee, but Oracle has been forced by the competition from Android to dramatically reduce the level of royalty that it gets for each device. And that's—that's harm when—if you've still got the sale but you make a lot less money on it, that's still harm.

Q. Now, we talked about the kinds of harm that Oracle has experienced in the market for the Java API packages.

What would be the consequence of permitting the kind of

* * *

[2013] I can make the same point by saying "meaning good faith or not," "evaluating good faith or not."

MS. SIMPSON: Your Honor, the *Harper* decision, I believe, disqualifies you from fair use if you're acting in bad faith. So I don't think that it is a far reach to include "bad faith" in the instruction at this point.

THE COURT: I remember that. It was an egregious set of facts. It was literally stolen in that case.

MR. BABER: That was the purloined unpublished manuscript, Your Honor.

Even in *Harper*, it didn't disqualify it, like end of story; we're not going to look at the factors. *Harper* went on and looked at all the factors.

MR. SILVERMAN: This instruction doesn't disqualify it. It says that it's one the factors.

But *Harper* says that fair use presupposes good faith and fair dealing.

THE COURT: All right. I'll consider it.

What else do you have on 27?

MR. BABER: The next thing I have, Your Honor, is in the second paragraph, which starts on line 24, which is the language you just added about recognized practices in the industry.

Instead of saying, "The extent to which Google relied on or contravened," rather than any concept of reliance, we think [2014] it should just be a more neutral, Did Google act in accordance with or contrary to any recognized practices in the industry?

THE COURT: Well, here's the problem with that: First, I want to go back and remind everyone that we—the Federal Circuit decision did not call out propriety of the defendant's conduct at all. That was not a factor that the Federal Circuit opinion called out.

However, Oracle pointed out that the *Harper & Row* decision by the Supreme Court did call that out as a factor, at least in that particular case. And I read the decision. It seems to me that a district judge can't ignore what the Supreme Court has said. So we put this factor in.

Well, once you decide that the copyright owner can show that the defendant did not act in good faith, the defendant is entitled to show they did act in good faith.

And, frankly, that's the main way in which all of your so-called custom evidence got in, was because anyone who's accused of bad faith has a right to get up on the stand and explain, no, I acted in good faith; and here's what my thinking was at the time, even if it doesn't rise to the level of a custom.

So I question—and I'm not saying I would rule on this, but I think it's a close call whether or not the evidence would be sufficient for a jury to conclude that there was a well-established custom of copying API declarations.

* * *

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF
CALIFORNIA

No. C 10-03561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

ORDER *IN LIMINE* RE ORACLE'S MOTION RE
DR. RODERIC CATTELL
(ECF NO. 1879)

Oracle's motion to exclude Dr. Roderic Cattell is **DENIED** except as follows:

Paragraph 31 of the report is excluded because what matters is re-implementing APIs *without a license or permission*, not just re-implementing APIs. Nor does he even say the copying was done without regard to whether or not a license to do so existed. Under Rule 403, this is more confusing than probative.

Paragraphs 46-47 are excluded as too inconsistent with the holding of the court of appeals.

Paragraphs 49-54 are excluded because they do not call out whether or not the re-implementations were

without permission or at least done without regard to whether or not a license to do so existed. Under Rule 403, this will be more confusing than probative. However, Paragraph 48 will be allowed since it specifically says "without permission from IBM."

As to opinions on direct examination, Cattell is limited to what is in his report minus the above paragraphs, and the fact that he added more opinions in his deposition does not allow those additional opinions on direct examination.

IT IS SO ORDERED.

Dated: May 12, 2016.

/s/ William Alsup
WILLIAM ALSUP
UNITED STATES DISTRICT JUDGE

472

Trial Exhibit 1

Android GPS

Key strategic decisions around Open Source

July 26, 2005

---

* * *

The Model

Google works closely with Carriers and OEMs to help incorporate its Open Source OS into handset designs.

- The Carriers benefit from the ability to quickly deploy differentiating features and applications.

- The OEMs benefit from the above, and a dramatic reduction in BOM costs by utilizing a robust, free consumer OS platform.
- Google benefits by having more control of the user experience and built-in Google apps.

Why Open Source[1]

To disrupt the closed and proprietary nature of the two dominant industry players: MSFT and Symbian

To provide Carriers and OEMs a non-threatening solution for cross-vendor compatibility

To eventually build a community force around Google handset APIs and applications

Which OSS Model?

- Build product

  A strategy where development happens internally, and once product has reached a certain level of stability it is released as open source.

- Community effort

  A strategy where development is a collaborative effort between internal Google resources and a larger more diverse community of external Open Source developers. Typically there is a publicly accessible source code repository with a informal check-in procedure.

---

[1] Our OSS license is a rider on a mainstream license such as Mozilla that requires licensee to maintain compatibility with Google APIs

JAVA

- Why Java?
  - Carriers require it
  - MSFT will never do it
  - Elegant tools story
  - Safe sandbox for 3rd party developers
  - Existing pool of developers and applications
  - Who pays? OEM pays sun a license, typically < .30 in volume
- Java+ Javascript/XML = Key differentiator

  Current scenario:
  - Developing a clean-room implementation of a JVM
  - Need coffee-cup logo for carrier certifications
  - Must take license from Sun
  - Cost isn't the issue, open source JVM is the issue

  Proposal:
  - Google/Android, with support from Tim Lindholm, negotiates the first OSS J2ME license with Sun
  - Outcome could reinforce our JVM development, or perhaps result in us convincing Sun to Open Source their MVM implementation.

* * *

Trial Exhibit 7

From: Tracey Cole. Sent: 10/11/2005 8:32 PM.
To: Jennifer; Andy Rubin.
Cc: LSA.
Bcc:
Subject: Re: Sun meeting.

thank you

———

From: Jenifer [mailto:jaustin@google.com]
Sent: Tuesday, October 11, 2005 7:39 PM
To: Andy Rubin
Cc: LSA; Tracey Cole
Subject: Re: Sun meeting

Larry is going to try to stop by the end of the meeting on Thurs, but he doesn't need to meet in advance to prep.

Deepest condolences,

Jenifer

On 10/11/05, Andy Rubin <arubin@google.com> wrote:

Begin forwarded message:

From: Andy Rubin < arubin@google.com>
Date: October 11, 2005 2:30:52 PM PDT
To: Larry Page < page@google.com>
Subject: Sun meeting

Larry,

We have been having discussions with Sun regarding Android's Open Source VM strategy. Alen Brenner, who owns the P&L for J2ME @ Sun is coming over to

basically tell us II believe) that Sun doesn't want to work with us. Alan's big concern is that by open sourcing our J2ME VM we will make licensing "enforceability" impossible for Sun—and he will lose revenue.

My proposal is that we take a license that specifically grants the right for us to Open Source our product. We'll pay Sun for the license and the TCK. Before we release our product to the open source community we'll make sure our JVM passes all TCK certification tests so that we don't create fragmentation. Before a product gets brough to market a manufacturer will have to be a Sun licensee, pay appropriate royalties, and pass the TCK again.

Sun has already permitted open source VM projects in non mobile areas—areas where they didn't have a well defined revenue stream. Apache is an example.

Android is building a Java OS. We are making Java central to our solution because a) Java, as a programming language, has some advantages because it's the #1 choice for mobile development b) There exists documentation and tools c) carriers require managed code d) Java has a suitable security framework

If Sun doesn't want to work with us, we have two options:

1) Abandon our work and adopt MSFT CLR VM and C# language

-or-

2) Do Java anyway and defend our decision, perhaps making enemies along the way

As you can see, the alternatives are sub-optimal, so I'd like you to stop in our Alan meeting and essentially

be the good cop. Let him know we love Sun, and want to find a way to do this Open Source thing.

I first looked to Eustace to help me, but he was out of office. Then Sergey—out also.

Thoughts?

- andy

Jenifer Austin Office of the Founders. Google, Inc. office 650-253-6327.

Trial Exhibit 10

From: Tim Lindholm Sent:8/612010 11:05 AM

To: Andy Rubin; benlee@google.com

Cc: Dan Grove; Tim Lindholm

Bcc:

Subject: Context for discussion: what we're really trying to do

Attorney Work Product

Google Confidential

Hi Andy,

This is a short pre-read for the call at 12:30. In Dan's earlier email we didn't give you a lot of context, looking for the visceral reaction that we got.

What we've actually been asked to do (by Larry and Sergei) is to investigate what technical alternatives exist to Java for Android and Chrome. We've been over a bunch of these, and think they all suck. We conclude that we need to negotiate a license for Java under the terms we need.

That said, Alan Eustace said that the threat of moving off Java hit Safra Katz hard. We think there is value in the negotiation to put forward our most credible alternative, the goal being to get better terms and price for Java.

It looks to us that Obj-C provides the most credible alternative in this context, which should not be confused with us thinking we should make the change. What we're looking for from you is the reasons why you hate this idea, whether you think it's a nonstarter

for negotiation purposes, and whether you think there's anything we've missed in our understanding of the option.

– Tim and Dan

Trial Exhibit 13

From: Brian Swetland Sent: 1/3/2006 1:31 PM
To: Mathias Agopian
Cc: fadden@google.com; arubin@google.com; joeo@google.com
Bcc:
Subject: Re: new java world

[Mathias Agopian <mathias@google.com>]

Has this decision been taken already or are we talking/arguing about it?

I think we're pretty set on it, but are still working on addressing issues people may have with it. The skia folks are being brought up to speed today. I unfortunately misremembered when you were going to be back (thought it was the beginning not the end of this week) and thought you would be around today to discuss things.

Brian

On Jan 2, 2006, at 11:07 PM, Brian Swetland wrote:

Reasons to shift to a primarily Java API

- single language massively simplifies the application development story: "you write android apps in java. native code is brought in as standalone modules (services) or as plugins to the runtime (components)"

- single language approach massively simplifies system development and reduces our development time. The universal multilanguage binding stuff is an awesome idea but is a crazy pile of work and risky.

- the tools story is much, much simpler. supporting gcc/etc cross compilers and other tools is a big pain (we have to do it for systems development but we can avoid passing that pain on to 99% of application developers).

- even on a system with processes / mmu / etc, java provides a nice safetynet and faster app development and debuggability. (this is based on experience developing hiptop—java saved us a pretty crazy amount of time).

- the negotiations with Sun are going far better than expected. A lot of the push for multi-language bindings was a result of Brian trying to work out a cover-our-ass setup for when Sun proves impossible to work with (he was perhaps a bit scarred by his danger experience).

- using java simplifies the "why did you invent a new api" story. The embedded java world has midp which we will have for compatibility with 'legacy' phone apps, but not much in the way of more powerful environments. We can fill this gap *and* avoid a lot of the "why didn't you just use gtk / qtopia / etc" questions.

- using java allows us to take advantage of a modern, garbage collected memory model for applications without having to worry about integration with C++ allocation, reference counting, etc. The same goes for synchronization.

- having one primary language environment allows us to focus all our language systems development energy on making the java world extremely fast and solid.

One language is okay, but why Java instead of C++, Intercal, etc?

- The nature of the cellular market is that we are *required* to have java due to carrier requirements, etc. Since we're sorta stick with that, we can provide two environments (java and native) like everyone else does, expending a lot more energy, or simplify and just provide a fantastic java environment.

- Java is more accessable than C++. There are more Java programmers. There is more standardization in tools and libraries. Debugging is much simpler (especially for people who are not total rockstars perhaps a lot of casual developers, etc).

- Java solves a lot of the portability issues C++ has. There is no fragile base class problem in the sense that it exists in C++. We can safely provide a modern object oriented api to third party developers without the scary ABI issues involved in C++. (exceptions are zero runtime cost if not throw, by design, in java. garbage collection is builtin and standard. etc. etc)

- Performance concerns? Yup, we need to do work to push the heavy lifting to native (but we're already doing that!) and have a care for performance and writing more of the system in java does make that a little harder. We solve this by making the java runtime very fast, moving what needs to be native native, and being smart about writing our java code. The folks from danger can explain this at length—shipping a *fast* java based system is totally doable, even on much slower hardware than we have.

- Java does have a big win of being much more compact code than native arm/thumb code.

What changes (not all that much):

- The biggest change is the view system becomes a java library. This does involve migrating some already written code, but in the end makes for much easier to use java apis if we don't restrict the design to the intersection of java and c++ features.

- Most of the low level, performance intense stuff (image libs, skia, audio engine, etc) stays just as it is and gets java wrappers as previously planned.

- The browser integration mostly moves ahead as planned. It is a native component represented in a java widget/view/whatever. A way of having it setup and control other widgets on top of (inside of?) itself will be required for forms support, etc.

- Joe's IDL tool is still used to generate java bindings for native components, but it lives in a simpler world. It is also used for building c++ and java interfaces to "services" which are the bigger building blocks of the system (data storage, addressbook, telephony, etc)

Components vs Services:

- Components plug in to the core system libraries and are exposed

* * *

Trial Exhibit 14

From: Andy Rubin. Sent: 1/13/2006 8:01 PM.
To: sergey@google.com.
Cc: Larry Page; LSA; Alan Eustace.
Bcc:
Subject: Sun Microsystems

Sergey,

When Android first arrived I did a GPS that explained the importance of Java in our solution.

Since then I've been working with Sun and pushing them to open source Java. Initially this was a foreign concept to them and took some educating. Now we're at a point where they have conceptually agreed to open java and additionally they desire to broaden the relationship and become a customer of the Android system and Google. Their desire is to create a "distribution" of the Android system ala Redhat. It will be an industry changing partnership. Sun is prepared to walk away from a $1OOM annual J2ME licensing business into an open source business model that we together crafted. This is a huge step for Sun, and very important for Android and Google.

Soon I will give a detailed presentation to EMG. I'm writing this e-mail tonight to give you a heads up that you may receive a phone call from Jonathan Schwartz. Alan Brenner (Sun VP) briefed him today and Jonathan was excited and immediately wanted to pickup the phone to call you. He doesn't know any of the details of the discussions, but apparently his team has armed him with some basic concepts of the Android project which you are familiar with.

I'm available via my cell phone if you need to reach me: [REDACTED]

-andy

Trial Exhibit 15

From: Andy Rubin. Sent: 2/5/2006 7:44 PM.
To: Rich Miner; Tim Lindholm.
Cc:
Bcc:
Subject: Fwd: EMG Deal Review Agenda and Slides - Feb 6, 2006.

We're on stage at 11 am. Probably get called in at 11:30

Begin forwarded message:

From: Gisel Hiscock <gisel@google.com>
Date: February 5, 2006 7:12:48 PM PST
To: "'Gisel Hiscock'" <gisel@google.com>, <emg@google.com>, "'Susan Wojcicki'" <susan@google.com>, <oea@google.com>, <joan@google.com>, "'Mark Fuchs'" <mfuchs@google.com>, "'Jim Marocco'" <jmarocco@google.com>, "'Stephane Panier'" <stephane@google.com>, "'Julio Pekarovic'" <julio@google.com>, <ostaff@google.com>, <EMGEA@google.com>, "'Joanna Shields'" <joannas@google.com>, "'Raif Jacobs'" <raif@google.com>, <nicolew@google.com>, <deal_review@google.com>, "'Larissa Fontaine'" <larissa@google.com>, <jdonovan@google.com>, <dnoonan@google.com>, "'Brad Chin'" <bchin@google.com>
Cc: <sang@google.com>, <kenc@google.com>, "'Jim Holden'" <jholden@google.com>, <sukhinder@google.com>, "'Sanjay Kapoor'" <sanjayk@google.com>, "'sacca'" <sacca@google.com>, "'Bill Brougher'" <billbr@google.com>, "'Kurt Abrahamson'" <kurt@google.com>, "'Andy Rubin'" <arubin@google.com>, <dalegre@google.com>,

<Crim@google.com>, <dnishar@google.com>, <tim@google.com>, "'Greg Badros'" <greg@google.com>, "'Gokul Rajaram'" <gokul@google.com>, <jaz@google.com>, <adamf@google.com>

Subject: RE: EMG Deal Review Agenda and Slides - Feb 6, 2006

Hi EMG -

Please fogive me. We actually will not be reviewing Empas/Paran tomorrow. Here is the updated agenda and new slides are attached.

1. AOL (Sanjay Kapoor) - slides will be sent tomorrow to a shortened distribution list.
2. Sun Handset Open Source (Andy Rubin)
3. Vail – M&A (Sean Dempsey) - tentative item/TBD

Meeting details below…

Thanks,
Gisel

From: Gisel Hiscock [mailto:gisel@google.com]

Sent: Sunday, February 05, 2006 6:24PM

To: emg@google.com; Susan Wojcicki; oea@google.com; joan@google.com; Mark Fuchs; Jim Marocco; Stephane Panier; Julio Pekarovic; ostaff@google.com; EMGEA@google.com; Joanna Shields; Raif Jacobs; nicolew@google.com; deal_review@google.com; Larissa Fontaine; jdonovan@google.com; dnoonan@google.com; Brad Chin

Cc: sang@google.com; kenc@google.com; Jim Holden; sukhinder@google.com; Sanjay Kapoor; sacca; Bill Brougher; Kurt Abrahamson; Andy Rubin;

dalegre@google.com; Crim@google.com; dnishar@google.com; tim@google.com; Greg Badros; Gokul Rajaram; jaz@google.com; adamf@google.com

Subject: [Deal_review] EMG Deal Review Agenda and Slides - Feb 6, 2006

Hi EMG -

We have 4 deals to review tomorrow at our EMG Deal Review meeting:

1. AOL (Sanjay Kapoor) - slides will be sent tomorrow to a shortened distribution list.

2. Empas/Paran Renewal – Korea Deal Update (Sang Kim)

* * *

Project Android

We are building the world's first Open Source handset solution with built-in Google applications

We are forming an alliance with interested parties to make this free platform the de facto standard for modern handsets

Partner Overview – Sun Microsystems

Who Are They?

- Products and services for network computing
  - Java dominates wireless industry
  - Carriers require Java in their terminal terminal specifications
- Statistics
  - Not profitable
  - $15.4B Market Cap
  - $11.6B revenue
  - $469M EBITDA
  - $2.45B cash
- Market Presence
  - 1B Java embedded handsets

  - 180 carrier deployments
- Size
  - Offices in 170 countries worldwide
  - 40k employees

Why Do the Deal?

- Critical to our open source handset strategy
- Dramatically accelerates our schedule
- Form an industry alliance to block MSFT
- Create value for wireless stakeholders

Status

- Discussions started as result of our last GPS
- Alliance consists of key players of the wireless industry, including handset OEMs and wireless carriers
- Sun becomes founding partner in alliance
- Companies engage in a co-development relationship

Proposed Deal Terms

Client

- Sun Microsystems

Term

- 3 years
- Co-development partnership

Proposal

- Sun makes Java Open Source as part of Android platform
- Companies work together to bring Android platform to market

Exclusivity

- Exclusive

Data Use / Restrictions

- See detailed slides

Fee

- $36M
- To be negotiated potential rev share on platform-enabled mobile ads

Attribution

- N/A

Termination / Other Issues

- N/A

* * *

Trial Exhibit 18

From: Greg Stein. Sent: 3/24/2006 5:01 PM.
To: Andy Rubin.
Cc:
Bcc:
Subject: Re: Confidential: the open J2ME project.

The APIs still allow compatible implementations. e.g. Apache Geronimo's implementation of the J2EE specification.

Which "shared part" are you referring to? Let's assume they'll be able to get the TCK. Then they build the J2ME. And it passes the TCK. What would they need to take from Sun?

And what can we do to get you, et al, talking with them? It would be great if we could trade info.

Thx,

-g

On 3/24/06, Andy Rubin <arubin@google.com> wrote:

Ha. Wish them luck. Java.lang apis are copyrighted. And sun gets to say who they license the tck to, and forces you to take the "shared part" which taints any cleanroom implementation.

-----Original Message-----

From: Greg Stein <gstein@google.com>
To: Andy Rubin arubin@google.com
Sent: Fri Mar 24 14:22:14 2006
Subject: Re: Confidential: the open J2ME project

Oh, they have a plan for that. The ability to call it Java(tm) is "simply" a matter of passing the J2ME TCK, as I understand it.

On 3/24/06, Andy Rubin <arubin@google.com> wrote: I don't see how you can open java without sun, since they own the brand and ip.

Happy to talk.

-----Original Message-----
From: Greg Stein <gstein@google.com>
To: Andy Rubin <arubin@google.com>
Sent: Fri Mar 24 14:08:29 2006
Subject: Confidential: the open J2ME project

Andy,

Chris DiBona said you're the right person to talk to about our J2ME plans with Sun. I've recently become aware of a similar effort to create an open source J2ME. The problem that I have right now is that I can't tell Google about that, and I can't tell them about our effort. IOW, without violating confidentiality *somewhere*, there is no way for me to make the two parties aware of each others' efforts.

I've asked them if I can at least mention a small amount to Google. Waiting on that reply. Is there any hint or small amount of into that I can give them so that we can open the door more?

* * *

Trial Exhibit 29

From: Andy Rubin. Sent: 3/24/2008 11:58 AM.
To: Dick Wall.
Cc: android-comms@google.com.
Bcc:
Subject: Re: [android-comms] Android presence at JavaOne.

1) Are we able to answer direct developer questions about Android at the booth?

Yes. One-on-one only, please.

2) Can we demonstrate the tooling, emulator, developer environment, etc?

Yes, one-on-one only please, where you know exactly who you are talking to. Please dont demonstrate to any sun employees or lawyers.

3) Assuming either of the above is possible, do we need to make it explicitly clear before talking to developers that Android does not use the JVM and has its own VM instead?

The truth is always the best approach.

4) Is the story of "you use Java source code, but libraries and VM differ from Java SE" still the right message to be carried?

"We use standard development tools such as Eclipse and IntelliJ and developers can use those tools to develop in the Java programming language"

5) Can we put any material out about Android at the booth, assuming such material has been scrutinized and approved first?

No.

On Mar 24, 2008, at 11:01 AM, Dick Wall wrote:

Hi Folks

I will be at JavaOne for the entire week and will be volunteering for several spots of booth duty. As a side effect of this I have some questions regarding Android.

I expect (in fact I am almost certain) that developers will visit the Google booth curious about Android. There are lots of different options if they do so, here are some of the questions I would like considered…

Are we able to answer direct developer questions about Android at the booth?

Can we demonstrate the tooling, emulator, developer environment, etc?

Assuming either of the above is possible, do we need to make it explicitly clear before talking to developers that Android does not use the JVM and has it's own VM instead?

Is the story of "you use Java source code, but libraries and VM differ from Java SE" still the right message to be carried?

Can we put any material out about Android at the booth, assuming such material has been scrutinized and approved first?

Finally, if none of the above is possible, what is our response for questions about Android at the booth? Just point them at android.com or press@google.com?

I ask this because JavaOne is the largest developer conference of its kind and will bring together a very large number of developers who are very suitable for

Android development. If there is any way we can hit them, it would be a good idea to try.

JavaOne is about 6 weeks away, but it would be great to get answers to these questions pretty soon, in case there is something we can do to get Android out there at JavaOne.

Cheers

Dick

Trial Exhibit 31

Google

Android 101

An introduction to Android and Android partnerships

Last updated: December 2008

* * *

Why Android?

- Does the world need another phone stack?
  - Android strategic proposition
- Why did Google invest in Android?
  - …and then give it away?
- What does Android mean for Partnerships?
  - Who do we partner with and how?

Does the world need another phone stack?

- The mobile world of tomorrow:
  - Today's phones are as powerful as yesterday's computers
  - Many people will never have a PC—only a mobile phone
- Held back by the mobile world of today:
  - Continued promise of a mobile Internet not yet delivered
  - Mobile ecosystem is broken

- Access to users is tightly controlled
- Broken relationship between OEMs and operators
- Need for more software expertise
- Limited power and freedom for developers
- Locked down devices and network access

The model is changing…

- A few years ago, mobile platforms were driven by OEM and carrier requirements
- Enter: **Microsoft**, **Apple**, **Blackberry** … and **Nokia** (s/w)
- Major shift to people who understand software, customer experience, application development
- **But the ecosystem is still broken and not open!**
  - Access to users still controlled
  - Other handset OEMs (50% of the market) don't have a solution
  - Service providers still at the mercy of those who control the access points

* * *

Trial Exhibit 134

From: Vineet Gupta - OEM Software Sales CTO/Worldwide SE Director Sent: 1/31/2006 2:45 PM
To: Andy Rubin
Cc:
Bcc:
Subject: Re: Urgent stats needed

Andy,

Handset data in pdf included - rest below..

Let me know if you need something more…

-Vineet

Andy Rubin wrote:

Vineet,

I'm proposing our deal to the execs today. Can you send me ASAP:

- Some market presence numbers for Sun java example: in xx handsets and written into xx carrier terminal specifications

- Some numbers that promote the JCP process: how many participants, etc.

The JCP has over 700 corporate and individual participants. Over 300 JSR's, industry wide participation in creating JSRs (including Google).

(jcp.org)

- How many employees at Sun

around 40K…

- How many offices, worldwide

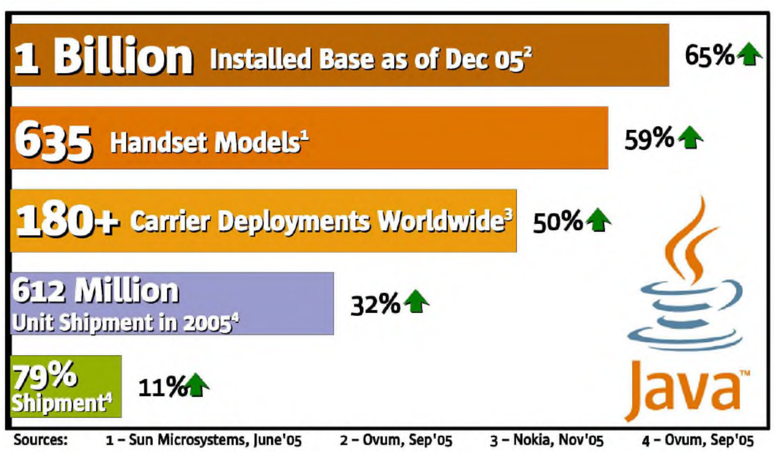
With offices located in over 170 countries worldwide, Sun has become a world leader in technology. Worldwide Headquarters is located in Santa Clara, California, USA.

These can be rough numbers.

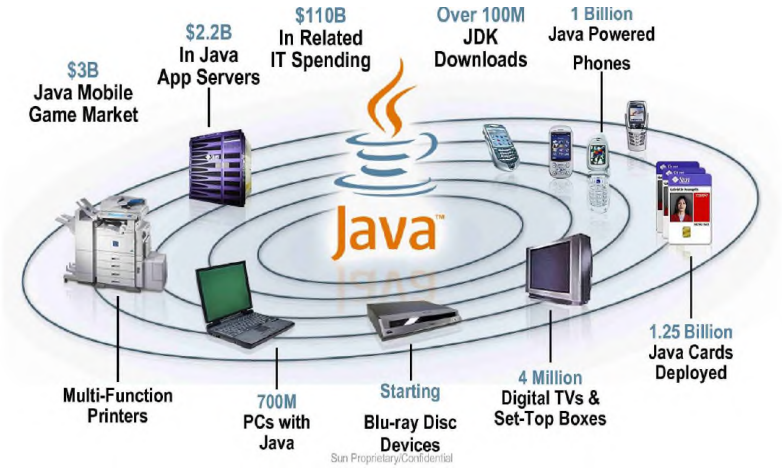
$$
Vineet Gupta
WorldWide Sr. Director
Chief Strategy/Technology officer
OEM Software Systems Engineering
SUN Microsystems
Vineet.Gupta@Sun.Com
(408)404-8950
$$

* * *

Java ME in Wireless
Phenomenal Momentum in 2004 – 2005



The Java™ Community Has Created Tremendous Opportunity



Trial Exhibit 158

From: andyt@google.com. Sent: 9/28/2006 8:05 PM
To: tangjianfeng@chinamobile.com
Cc: billwli@google.com; aruubin@google.com
Bcc:
Subject: Materials on Google Open Handset OS
Dir. Tang,

Pls see attached for the materials on our open handset OS.

Let me know if you need more info,

Thanks!

Andy

* * *

A complete stack is the way to accelerate adoption

Fact: Industry noise around Linux is at its all-time high. Still, no one is offering a complete platform in an open way—instead, we find people using open source as a marketing advantage, and offer only certain layers of the stack. Examples:

- Access/PalmSource: Using Linux to build a closed system
- Motorola/Samsung/NTT/Vodafone-Japan: Trying to specify apps environment.

  Where are the software companies?
- TrollTech: Green Phone. NOT OPEN.
- MLI, LiPS: Specification only, no implementation.

Strategy: Open Source the entire stack only after the first devices show up in the market. Send a strong signal to the industry that they now have everything they need to build devices as-good-as or better than the ones we just released.

Supporting Java is the best way to harness developers

Fact: Linux fragmentation threatens value. Tools and new app frameworks are biggest hurdles. 6M Java developers worldwide. Tools and documentation exist to support app development without the need to create a large developer services organization. There exist many legacy Java applications. The wireless industry has adopted Java, and the carries require its support.

Strategy: Leverage Java for its existing base of developers. Build a useful app framework (not J2ME). Support J2ME apps in compatibility mode. Provide an optimized JVM (Dalvik). Integrate class libraries and other technology from Skelmir acquisition to accelerate effort.

* * *

Trial Exhibit 205

From: Scott McNealy Sent: 2/8/2006 5:09 AM
To: Eric Schmidt
Cc: scott.g.mcnealy@Sun.com; Andy Rubin; Jonathan Schwartz; Greg Papadopoulos
Bcc:
Subject: Re: Potential Sun Google partnership in the Mobile Java and OS Space

Thanks for the note. Jonathan and the team are on top of this—I'm worried about how we're going to replace the revenue this is likely going to submarine. I'm very supportive of driving a completely open phone stack, and even taking risk with Java to get there, but I just need to understand the economics.

But we're obviously supportive in helping to fuel the market.

Scott

PS Has you team had a chance to try out the new T2000 with 32 hw threads yet?

Eric Schmidt wrote:

Scott.. I'm in a product review and we are looking at a very interesting partnership proposal with Sun. Basically, Andy Rubin runs our mobile os/search engineering group; he is talking with Alan Brenner, VP Consumer & Mobile Systems Group of Sun.

Google has engaged with Sun's Java team in an effort to form an alliance around our Open Handset Platform. It is an opportunity for our two Companies to work together to define the de-facto standard software stack for mobile phones. Google has adopted a completely open source model as a way to solve some

intricate distribution dependencies. It makes sense to me that Sun and Google should do this together; can you check and hopefully dedicate the resources necessary to move this ahead at an accelerated pace. I wanted to make sure you know I will do the same on my side.

Anyway, this is very exciting and hopefully a good idea for both of us!

Thanks and take care.. Eric

Trial Exhibit 215

From: Chris Desalvo Sent: 6/1/2006 4:10 PM
To: Rubin Andy
Cc:
Bcc:
Subject: Java class libraries

With talks with Sun broken off where does that leave us regarding Java class libraries? Ours are half-ass at best. We need another half of an ass.

-chris

Trial Exhibit 370

Mobile Strategy Summit - Notes
November 4-5, 2010

Day 1

Opening [Henrique]

- Mobile is certainly technology disruptor; if we miss the "mobile window", we'll be out of business in 10 years
- Key is to replicate our historical success in desktop Search/AFS on mobile
- Achieve scale by leveraging mobile sales force as a specialist team—specialist team empowers the broader sales force to sell
  - The search product and the pods will open doors and provide access to huge advertiser base
- Sales should feel comfortable lifting revenue forecasts, be aggressive w/ their projections

Kick-Off [Karim]

- Strategic offsites on quarterly basis
  - Remind what we're about, where we're going, next products
- There will be failures along the way, so we can't be afraid of making them. We need to be realistic about failures on our way to >$5B
- Empower your teams to make lots of executive decisions. Business moving, growing, and changing quickly.
- Will adopt metrics-based approach to driving and evaluating the business
- Strive to achieve operational excellence. Strategy is important, operational excellence is critical. Be prepared to be scrappy at times.

- Be bold! Don't look for incremental growth. Chase step-function increases.
- Objectives of this summit
  - Stay engaged in discussions
  - Voice disagreements
  - Manage by consensus—even if you disagree, we all ask that you support the team's final decision
- Mobile will be ubiquitous—will be the most important item for everyone in the future (payment, interaction, gaming)
- Collaboration is key—mobile org must be entirely embedded into broader Google org
  - Regional mobile leaders need to feel and operate like they are part of the regional sales teams under Dennis, Philipp, Marco, and Daniel
- Surpassing $5B runrate
  - Let's think about blowing out the 2.5B in 2011 and 5B in 2012 run-rate goals
- Marketing Investment Framework
  - may evolve over next few months
  - tight timeframe for expanding into markets
- Countries need to own mobile thought leadership
- Focus on internal communications: w/ each other, w/ pods, and w/ prod/eng
- Investment case
  - Huge opportunity in Japan. Market shifting towards HED and display.
  - High potential markets (display in China; redo Google search strategy in Korea)
  - Emerging markets

Product Review [Paul F]

- 2/3 of mobile revenue comes from Search/AFS/Distribution; 14% from AdMob; GMM ~0%
- HE quickly outpacing WAP
- CTC almost at $100M RR
- 4 Challenges
  - Display ad revenue optimization
    - Mobile technology differences (smaller screens, html5/flash) means many Google products don't work well.
    - Developers have similar problems to desktop (direct sales, yield mgmt, interest in rich media/brand)
    - Developers have different problems from desktop (ad mediation, revenue models, app promo)
    - **AI: Paul/Jason - What product suite is right for app developers? How do we execute / deeper dive?**
  - Mobile eCommerce unsolved
    - Commerce on mobile (web/apps) growing quickly—Amazon + eBay >$2.5B, mobile transactions still difficult for consumers
    - Apps are creating new parallel internet
      - Open Table thinks that its traffic will move almost entirely to mobile apps in 3 yrs (away from desktop websites)
      - Paul building track conversions of activities deep w/in apps (tracks what happens after install); e.g., purchasing an item deep w/in an app
      - Targeting and ad selection problem is different

        - Position: we should focus on ads, not commerce. We're trying to make transactions as frictionless as possible
    - Real world commerce is unsolved
        - Holy grail = bringing targeting, interactivity, measurement, accountability
    - Mobile will completely connect eCommerce and real world commerce
        - Bricks-to-clicks as likey as clicks-to-bricks
- Approach to display
    - End goal: mobile as first class citizen on Google platform
    - Full speed ahead on integration to get benefit of core platform—integration should focus on simplifying for advertisers, pubs, and sales
        - Pub front end will be XFP (while accounting for AdSense)
        - Adv front end will be AdWords
            - Key features from AdMob: targeting, format, measurement, reservations, workflow (all in phase 2)

* * *

Trial Exhibit 610.1

Java 2 Platform Standard Edition Development Kit 5.0

Specification: JAVA 2 PLATFORM STANDARD EDITION DEVELOPMENT KIT 5.0 Specification ("Specification")

Status: Final Release
Release: August 25, 2004

include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space other than those required/authorized by the Specification or Specifications being implemented; and (iii) passes the TCK (including satisfying the requirements of the applicable TCK Users Guide) for such Specification. The foregoing license is expressly conditioned on your not acting outside its scope. No license is granted hereunder for any other purpose.

You need not include limitations (i)-(iii) from the previous paragraph or any other particular "pass through" requirements in any license You grant concerning the use of your Independent Implementation or products derived from it. However, except with respect to implementations of the Specification (and products derived from them) that satisfy limitations (i)-(iii) from the previous paragraph, You may neither: (a) grant or otherwise pass through to your licensees any licenses under Sun's applicable intellectual property rights; nor (b) authorize your licensees to make any claims concerning their implementation's compliance with the Spec in question.

For the purposes of this Agreement: "Independent Implementation" shall mean an implementation of the Specification that neither derives from any of Sun's source code or binary code materials nor, except with an appropriate and separate license from Sun, includes any of Sun's source code or binary code materials; and "Licensor Name Space" shall mean the public class or interface declarations whose names begin with "java", "javax", "com.sun" or their equivalents in any subsequent naming convention adopted by Sun

through the Java Community Process, or any recognized successors or replacements thereof.

This Agreement will terminate immediately without notice from Sun if you fail to comply with any material provision of or act outside the scope of the licenses granted above.

TRADEMARKS: No right, title, or interest in or to any trademarks, service marks, or trade names of Sun, Sun's licensors, Specification Lead or the Specification Lead's licensors is granted hereunder. Sun, Sun Microsystems, the Sun logo, Java, J2SE, J2EE, J2ME, Java Compatible, the Java Compatible Logo, and the Java Coffee Cup logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

DISCLAIMER OF WARRANTIES: THE SPECIFICATION IS PROVIDED "AS IS". SUN MAKES NO REPRESENTATIONS OR WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE OR THAT ANY PRACTICE OR IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADE SECRETS OR OTHER RIGHTS. This document does not represent any commitment to release or implement any portion of the Specification in any product.

THE SPECIFICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED

TO THE INFORMATION THEREIN; THESE CHANGES WILL BE INCORPORATED INTO NEW VERSIONS OF THE SPECIFICATION, IF ANY. SUN MAY MAKE IMPROVEMENTS AND/OR CHANGES TO THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THE SPECIFICATION AT ANY TIME. Any use of such changes in the Specification will be governed by the then-current license for the applicable version of the Specification.

LIMITATION OF LIABILITY: TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO ANY FURNISHING, PRACTICING, MODIFYING OR ANY USE OF THE SPECIFICATION, EVEN IF SUN AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You will indemnify, hold harmless, and defend Sun and its licensors from any claims arising or resulting from: (i) your use of the Specification; (ii) the use or distribution of your Java application, applet and/or clean room implementation; and/or (iii) any claims that later versions or releases of any Specification furnished to you are incompatible with the Specification provided to you under this license.

RESTRICTED RIGHTS LEGEND: U.S. Government: If this Specification is being acquired by or on behalf

of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in the Specification and accompanying documentation shall be only as set forth in this license; this is in accordance with 48 C.F.R. 227.7201 through 227.7202-4 (for Department of Defense (DoD) acquisitions) and with 48 C.F.R. 2.101 and 12.212 (for non-DoD acquisitions).

REPORT: You may wish to report any ambiguities, inconsistencies or inaccuracies you may find in connection with your use of the Specification ("Feedback"). To the extent that you provide Sun with any Feedback, you hereby: (i) agree that such Feedback is provided on a non-proprietary and non-confidential basis, and (ii) grant Sun a perpetual, non-exclusive, worldwide, fully paid-up, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation the Feedback for any purpose related to the Specification and future versions, implementations, and test suites thereof.

GENERAL TERMS: Any action related to this Agreement will be governed by California law and controlling U.S. federal law. The U.N. Convention for the International Sale of Goods and the choice of law rules of any jurisdiction will not apply.

The Specification is subject to U.S. export control laws and may be subject to export or import regulations in other countries. Licensee agrees to comply strictly with all such laws and regulations and acknowledges that it has the responsibility to obtain such licenses to export, re-export or import as may be required after delivery to Licensee.

Neither party may assign or otherwise transfer any of its rights or obligations under this Agreement, without the prior written consent of the other party, except that Sun may assign this Agreement to an affiliated company.

This Agreement is the parties' entire agreement relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, conditions, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification to this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

(Sun.CfcsSpec.license.11.14.2003)

Trial Exhibit 877

Joshua Bloch: Bumper-Sticker API Design

Posted by Joshua Bloch on Sep 22, 2008
Community Java, .NET, Architecture, Ruby
Topics Programming

My conference session *How to Design a Good API and Why it Matters* has always drawn large crowds; on InfoQ was the third most viewed content last year. When I presented this session as an invited talk at OOPSLA 2006, I was given the opportunity to write an abstract for the proceedings. In place of an ordinary abstract I decided to try something a bit unusual: I distilled the essence of the talk down to a modest collection of pithy maxims, in the spirit of Jon Bentley's classic Bumper-Sticker Computer Science, Item 6 in his excellent book, *More Programming Pearls: Confessions of a Coder* (Addison-Wesley, 1988).

It is my hope that these maxims provide a concise summary of the key points of API design, in easily digestible form:

All programmers are API designers. Good programs are modular, and intermodular boundaries define APIs. Good modules get reused.

APIs can be among your greatest assets or liabilities. Good APIs create long-term customers; bad ones create long-term support nightmares.

Public APIs, like diamonds, are forever. You have one chance to get it right so give it your best.

APIs should be easy to use and hard to misuse. It should be easy to do simple things; possible to do

complex things; and impossible, or at least difficult, to do wrong things.

APIs should be self-documenting: It should rarely require documentation to read code written to a good API. In fact, it should rarely require documentation to write it.

When designing an API, first gather requirements—with a healthy degree of skepticism. People often provide solutions; it's your job to ferret out the underlying problems and find the best solutions.

Structure requirements as use-cases: they are the yardstick against which you'll measure your API.

Early drafts of APIs should be short, typically one page with class and method signatures and one-line descriptions. This makes it easy to restructure the API when you don't get it right the first time.

Code the use-cases against your API before you implement it, even before you specify it properly. This will save you from implementing, or even specifying, a fundamentally broken API.

Maintain the code for uses-cases as the API evolves. Not only will this protect you from rude surprises, but the resulting code will become the examples for the API, the basis for tutorials and tests.

Example code should be exemplary. If an API is used widely, its examples will be the archetypes for thousands of programs. Any mistakes will come back to haunt you a thousand fold.

You can't please everyone so aim to displease everyone equally. Most APIs are overconstrained.

Expect API-design mistakes due to failures of imagination. You can't reasonably hope to imagine everything that everyone will do with an API, or how it will interact with every other part of a system.

API design is not a solitary activity. Show your design to as many people as you can, and take their feedback seriously. Possibilities that elude your imagination may be clear to others.

Avoid fixed limits on input sizes. They limit usefulness and hasten obsolescence.

Names matter. Strive for intelligibility, consistency, and symmetry. Every API is a little language, and people must learn to read and write it. If you get an API right, code will read like prose.

If it's hard to find good names, go back to the drawing board. Don't be afraid to split or merge an API, or embed it in a more general setting. If names start falling into place, you're on the right track.

When in doubt, leave it out. If there is a fundamental theorem of API design, this is it. It applies equally to functionality, classes, methods, and parameters. Every facet of an API should be as small as possible, but no smaller. You can always add things later, but you can't take them away. Minimizing conceptual weight is more important than class- or method-count.

Keep APIs free of implementations details. They confuse users and inhibit the flexibility to evolve. It isn't always obvious what's an implementation detail: **Be wary of overspecification**.

Minimize mutability. Immutable objects are simple, thread-safe, and freely sharable.

Documentation matters. No matter how good an API, it won't get used without good documentation. Document every exported API element: every class, method, field, and parameter.

Consider the performance consequences of API design decisions, but don't warp an API to achieve performance gains. Luckily, good APIs typically lend themselves to fast implementations.

When in Rome, do as the Romans do. APIs must coexist peacefully with the platform, so do what is customary. It is almost always wrong to â€œtransliterateâ€ an API from one platform to another.

Minimize accessibility; when in doubt, make it private. This simplifies APIs and reduces coupling.

Subclass only if you can say with a straight face that every instance of the subclass is an instance of the superclass. Exposed classes should never subclass just to reuse implementation code.

Design and document for inheritance or else prohibit it. This documentation takes the form of selfuse patterns: how methods in a class use one another. Without it, safe subclassing is impossible.

Don't make the client do anything the library could do. Violating this rule leads to boilerplate code in the client, which is annoying and error-prone.

Obey the principle of least astonishment. Every method should do the least surprising thing it could, given its name. If a method doesn't do what users think it will, bugs will result.

Fail fast. The sooner you report a bug, the less damage it will do. Compile-time is best. If you must fail at run-time, do it as soon as possible.

Provide programmatic access to all data available in string form. Otherwise, programmers will be forced to parse strings, which is painful. Worse, the string forms will turn into de facto APIs.

Overload with care. If the behaviors of two methods differ, it's better to give them different names.

Use the right data type for the job. For example, don't use string if there is a more appropriate type.

Use consistent parameter ordering across methods. Otherwise, programmers will get it backwards.

Avoid long parameter lists, especially those with multiple consecutive parameters of the same type.

Avoid return values that demand exceptional processing. Clients will forget to write the special-case code, leading to bugs. For example, return zero-length arrays or collections rather than nulls.

Throw exceptions only to indicate exceptional conditions. Otherwise, clients will be forced to use exceptions for normal flow control, leading to programs that are hard to read, buggy, or slow.

Throw unchecked exceptions unless clients can realistically recover from the failure.

API design is an art, not a science. Strive for beauty, and trust your gut. Do not adhere slavishly to the above heuristics, but violate them only infrequently and with good reason.

Watch Presentation: How to Design a Good API & Why it Matters

Joshua Bloch is Chief Java Architect at Google, author of *Effective Java, Second Edition* (Addison-Wesley, 2008), and coauthor of *Java Puzzlers: Traps, Pitfalls, and Corner Cases* (Addison-Wesley, 2005) and *Java Concurrency in Practice*. He was a Distinguished Engineer at Sun Microsystems, where he led the design and implementation of numerous Java platform features including JDK 5.0 language enhancements and the Java Collections Framework. He holds a Ph.D. from Carnegie-Mellon and a B.S from Columbia.

Trial Exhibit 951

Google Inc. GOOG
Q3 2010 Earnings Call Transcript

Executives

- Nikesh Arora : President, Global Sales Operations and Business Development
- Jonathan Rosenberg : SVP, Product Management
- Patrick Pichette : SVP and CFO
- Eric Schmidt : Chairman and CEO
- Jane C. Penner : IR Analysts
- Jeetil Patel : Deutsche Bank Securities
- Spencer Wang : Credit Suisse
- Mark Mahaney : Citi
- Douglas Anmuth : Barclays Capital
- James Mitchell : Goldman Sachs
- Jordan Rohan : Stifel Nicolaus
- Imran Khan : JPMorgan
- Jason Helfstein : Oppenheimer & Co
- Marianne Wolk : Susquehanna
- Benjamin Schachter : Macquarie
- Ross Sandler : RBC Capital Markets
- Jason Maynard : Wells Fargo
- Sandeep Aggarwal : Caris & Co
- Youssef Squali : Jefferies
- Justin Post : Merrill Lynch
- Mark May : Needham & Company
- Brian Pitz : UBS

Transcript Call Date 10/14/2010

Operator: Good day and welcome everyone to the Google, Inc. Third Quarter 2010 Earnings Conference Call. Today's call is being recorded. At this time, I

would like to turn the call over to Ms. Jane Penner, Senior Manager, Investor Relations. Please go ahead, ma'am.

Jane C. Penner - IR: Good afternoon, everyone, and welcome to today's third quarter 2010 earnings conference call. With us are Patrick Pichette, Chief Financial Officer; Jonathan Rosenberg, Senior Vice President, Product Management and Nikesh Arora, President, Global Sales Operations and Business Development.

First, Jonathan and Patrick will provide us with their thoughts on the quarter. Then Nikesh will join Patrick and Jonathan to answer your questions. Also, as you know, we recently began distributing our earnings release exclusively through our Investor Relations website located at investor.google.com. So, please refer to our IR website for earnings releases as well as supplementary slides that accompany the call. This call is also being webcast from investor.google.com. A replay of the call will be available on our website in a few hours.

Now, let me quickly cover the Safe Harbor. Some of the statements we make today may be considered forward-looking, including statements regarding Google's future and investments in our long-term growth and innovation, the expected performance of our business and our expected level of capital expenditures. These statements involve a number of risks and uncertainties that could cause actual results to differ

* * *

feeds from real-time information providers. We have real-time index, which of course, is very successful for us and you could see that whenever anything interesting happens, it is already right there at Google. For example, we use Twitter as a real time source of information. So if you search for almost anything, you will see that Twitter feeds now is part of universal search. Could you repeat the second part of your question for me?

Imran Khan - JPMorgan: Second one was, Eric, on the mobile front, right. We are seeing explosion on application on mobile platforms, so does that impact your search volume as people go directly to the vendors through the app? This seems like Amazon talked about $1 billion sell through coming from mobile devices, how does that impact Google's business long-term?

Eric Schmidt - Chairman and CEO: Doesn't seem to. This is one of those sort of worry word questions that we get all the time, that the success of one thing could that impinge on something else and in fact, the rising tide lifts and all those. I would say that again, what I hear is this sort of presumption that it's a zero-sum game and that one wins and another one loses. What's really happening is that all of the companies that are driving the web and web applications are all doing really well. People are moving from offline to online and in the course of doing that they are using these systems more. They are searching more, using apps more, et cetera. Now, from our perspective, you have this phenomenal success of Android, which is well past anything that I had ever hoped for, and looks like it's on its way to being a huge, huge success with a number of partners, number of devices, an

open model for access, lots and lots of innovation, more dynamic, more competition than any other part of the platform. There are up to 90,000 applications on Android growing very, very fast. Those applications, of course, have search services inside of them. So, we don't see them as a negative, we see both as very strongly positive.

Patrick Pichette - SVP and CFO: Let me go back then to Imran to the tax question. We did get a one-time benefit this quarter on the tax side and it is related. As the statutes expire for 2005 and 2006, all of our taxes are now closed and in doing so then we had an opportunity to reverse a set of provisions we had taken that we ultimately didn't have to take. So, you can see the tax rate this quarter is bit of anomaly again.

Operator: Justin Post, Merrill Lynch.

Justin Post - Merrill Lynch: Just when you think about Android as an operating system, how does that proprietary to Google when you think about your search services. Does that give you an advantage over other phones for some of your services and does the phone operate better when you are using Google Services? Second, I think you were quoted in a article saying, maybe someday Google can make $10 per phone, would you see that as mostly advertising and is that number right, something you did say?

Eric Schmidt - Chairman and CEO: The latter when I made out of thin air. So, we don't really have a notion of exactly what it is, but it's probably pretty big. So one way to think about Android is that it's probably the largest single platform play available in the market today, because it's a platform for

computation for location, for everything that you could do with the new and most popular set of computing devices that are emerging. That market is larger than the PC market, and the Tablet market is a small component of it, but an important part of it. So if you think as Mobile as platform as phone plus Tablet plus all the other things, we hope to become the leading platform in that space, and we are doing it with open source approach. So, in the open source approach that means, we give the software away, which is always paradoxical, people say how do you make money from that. Well let's start with the fact that the evidence we have is that the people who use Android, search twice as much as everything else. So, clearly there is more revenue associated with those searches. Other thing, of course, is if they are using Android systems, the revenue that we share and the searches are shared with the operator, but not with anybody else. So, again, it's more lucrative. So, not only is there more searches, and there's more ads, but it's also more lucrative. So, on that basis alone, Android is hugely profitable and we maintain the anti-fragmentation and other things by a series of contracts around their store and so forth and so on. So, Android is likely be financially successful to Google, without even any of the applications that are possible. So, Patrick calls up and says, okay, what else can you do for us, and the answer, of course, is that we can layer on value-added service, is usually how you get to the $10 and the value-added services could be of any kind. Our primary purpose right now is building this open platform. Google had chose to make it that on open systems and open platforms and open web; that served

us well so far and it looks like it is going to work really well on Android.

Operator: Mark Mahaney, Citigroup.

Mark Mahaney - Citi: Two questions, sequentially, is there something that you have been able to put in place that gives you confidence that that will continue going forward? Just on the Mobile revenue opportunity, are the results strong enough from your perspective in terms of dollars and growth such that you'll stay with an indirect monetization approach towards Android or are you going to keep the door open and potentially charge per operating system as a share of applications in the future?

Patrick Pichette - SVP and CFO: On cost per employee look, its just another reflection of, I wouldn't read anything kind of forward-looking into our results, except that, its just another good example of how we are – I have talked earlier about generous but frugal, we're investing, but people shouldn't confuse the fact that we're investing and we're investing aggressively where we really see fantastic opportunities from being wasteful. We're just not a wasteful company, and so in that sense, it does look as a good signal and we'll continue to do so. On the Mobile, maybe Jonathan or Nikesh can give us a indication of it?

Nikesh Arora - President, Global Sales Operations and Business Development: I think just following up on what Eric said earlier, we are very, very keen to build this Ecosystem and I think Jonathan's disclosure on the fact that we're on $1 billion run rate in Mobile, is testament to the fact that, now we have a revenue model, which we are very excited about, and that revenue model sort of proves to us that,

roughly the revenues are split between our search efforts, our display efforts and our application efforts. We are able to play across all those three spaces with our mobile monetization efforts, and the more people who use smartphones, the more people who are able to access (throughout) on their devices, the more we see the trend that people are going to search in them, they're going to give us opportunities to put display advertising on them. So, we see no reason to change our monetization model. We think the current approach to Android drives more users and more as usage and drives the Ecosystem.

Patrick Pichette - SVP and CFO: Nikesh, I think you've argued that display will become a very large component of the mobile revenue, because of the success that we're seeing in our mobile users and the hockey stick they are in.

Nikesh Arora - President, Global Sales Operations and Business Development: Exactly.

Operator: Douglas Anmuth, Barclays Capital.

Douglas Anmuth - Barclays Capital: Two things, first on Display. Can you give us some context in terms of breaking out YouTube AdSense for content in the Ad Exchange? Secondly, what's your view on other potentially competitive Android app stores that are out there?

Patrick Pichette - SVP and CFO: So, I'll answer the first and then I'll give Jonathan to talk about the Android marketplaces. So, on Display, we just don't break it down. So, we will not give the details. What we wanted to give today with the numbers we're sharing is a sense of scale and trajectory, and that's really

what we wanted to share. So, we unfortunately won't give any more details on that. On the Android competitive stores, Eric, maybe you can give us more perspective on it?

Eric Schmidt - Chairman and CEO: The goal of the stores is to make money for the people who are writing the software in their applications, and it's not a revenue goal for Google. So, there certainly will be multiple stores, they will certainly be the key one from us, and we think it's a net win for everybody, but it's not a primary focus from Google from a revenue perspective. It's really for the developers.

Operator: Brian Pitz, UBS.

Brian Pitz - UBS: Would you provide us with a relative idea of how the difference between average CPCs and clicker rates are basically on mobile versus the PC, now that you have a large enough number of devices in the market? Secondly, if there is a gap which I imagine there is, can you close that gap longer term between the two?

Jonathan Rosenberg - SVP, Product Management: Nikesh, can maybe give you more of a customer-based perspective. I think that some of you know we've recently started smart pricing on the mobile devices, and it is the case that the CPCs on the mobile devices are good bit lower. It's primarily because there isn't the measurement, there isn't as much of consummation of a transaction on the mobile devices. People don't have their credit cards in them. It's harder to type into them. So the mobile rates remain relatively lower. As payment platforms get built into the mobile devices and as people are more likely to actually complete the transaction, I think you'll see

those things go up substantially. I think it's also the case that on devices like the iPad, the kind of activity looks a little bit more like it does on a PC, primarily because people have a larger window, a bigger browser and they are also more able to input information. Nikesh?

Nikesh Arora - President, Global Sales Operations and Business Development: I think the only thing I have to add to that is there are some formats, which we started to introduce, which are driving a better monetization on the mobile sites, formats like Click to Call and hyper-local, because people are searching in their mobile devices where they want to then make a phone call or they are searching on their devices when they are looking for something in a very local context and there we're beginning to see sort of better CPMs and better monetization. Generally, we think that's where the trend is, that's where we're going to see more and more monetization, and clearly, we are seeing monetization in the application side of the mobile, because with the AdMob sort of team that we have and all the advertisers who want to be part of the application, the applications are becoming a big share of people's mobile usage.

* * *

Trial Exhibit 1056

From: Jonathan Schwartz <jis@sun.com>
Sent: Wed Mar 26 2008 17:20:12 PDT
To: Marten Mickos <marten@mysql.com>
CC: Greg Papadopoulos <greg.papadopoulos@sun.com>
Subject: Re: no doubt you saw…
Attachments:

Importance: Normal
Priority: Normal
Sensitivity: None

I so totally agree with you. We all do.

They also take Java for Android, without attribution or contribution.

This is why I love scroogle :-)

http://www.scroogle.org/cgi-bin/scraper.htm

On Mar 26, 2008, at 1:13 PM, Marten Mickos wrote:

Jonathan,

It's funny with Google. They take (without paying):
* the FOSS code of 10 million developers
* the web contents of 100 million websites
* the searches of 1,000 million web users

and add some magic of their own, after which they sell ads on this to some 0.1 million companies. And everyone is happy.

///mgm

Jonathan Schwartz wrote:

…was with my Google buddy over the weekend, and we got to talking about licenses. He made some pretty

interesting comments about their internal (as communicated by senior mgrs) view of licenses. They hate GPL, they like Apache, and they love BSD. Just like Microsoft…

On Mar 25, 2008, at 8:12 AM, Marten Mickos wrote:

Jonathan,

Yep. Expected.

They have spent $27m investing in the EnterpriseDB brand and now they switch to "Postgres Plus".

The list of Postgres attempts is getting long:
* Great Bridge in the early 2000s
* Red Hat had "Red Hat Database"
* Progress had UltraSQL
* CommandPrompt had MammothSQL
* Pervasive sold Postgres support for a while
* EnterpriseDB has Postgres Plus
* in Canada there is a PosgreSQL Inc
* and probably some more

///mgm

Jonathan Schwartz wrote:

http://www.news.com/newsblog/8301-10784_3-9901973-7.html from where I sit… the weird thing about Postgres is that the copyright isn't owned by a company, it's owned by a collective - via a BSD license that seems to spawn lots of small companies, but no center of mass… allowing us to say "we suppor Postgres," and putting folks like EnterpriseDB into a position of always having to explain who they are…

--
Marten Mickos, SVP, Database Group, Sun Microsystems

--
Marten Mickos, SVP, Database Group, Sun Microsystems

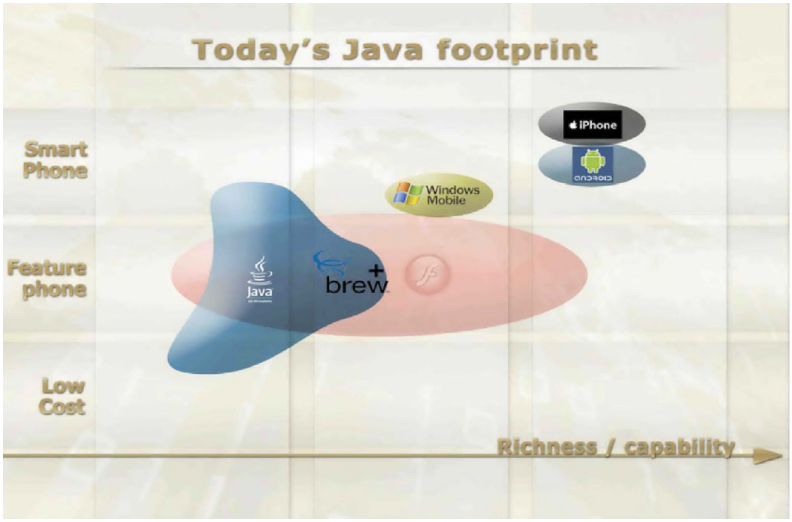
Trial Exhibit 2052

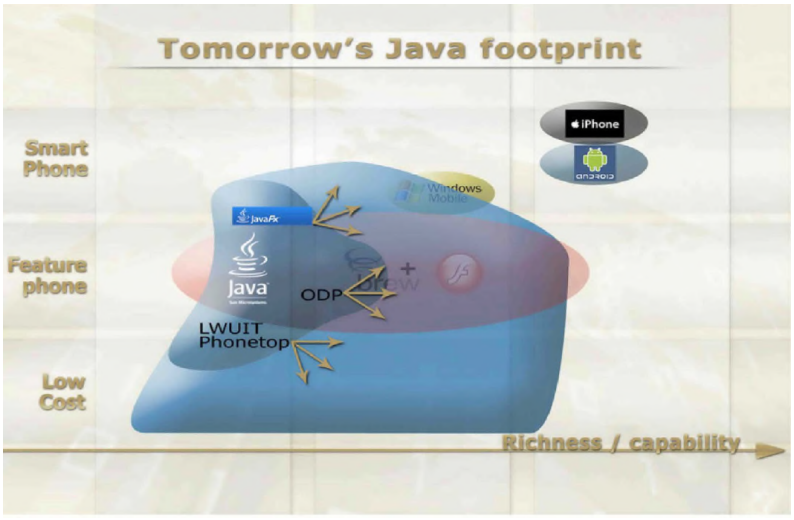
Java in Wireless Business Review

Craig Gering
Sr. Director
Mobile & Embedded

March 16, 2009

* * *

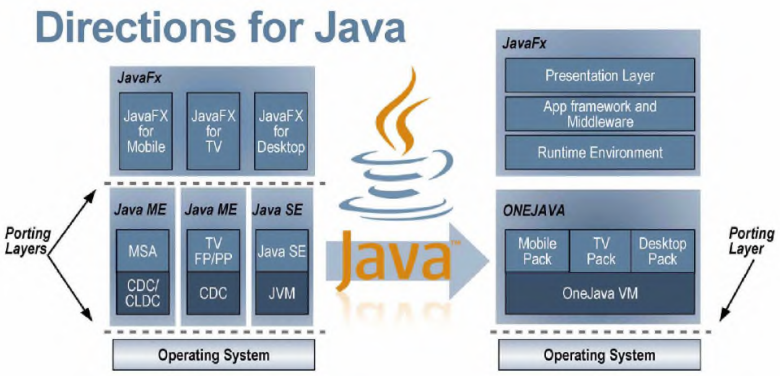




OneJava Vision

- One modular Java across all screens of your life
  - > Scalable footprint for an ever increasing range of devices
  - > Support for common JavaFX functionality on all screens

- Best of class universal language
  - > Leverage the latest Java language set
  - > Enabling 6+ million developers to reach outside the desktop
  - > Creating an even richer pool of tools



Current Java Platforms

- Different versions of Java language
- Incompatible security models
- Fragmented JSR sets

OneJava

- Unified language features
- VM selection based on footprint and performance considerations
- Base packs (mobile, TV, SE) focus on feature sets

* * *

Trial Exhibit 2368

From: Jonathan Schwartz
To: John Fowler
Sent: 11/7/2007 1:45:58 AM
Subject: Re: JAVA

I have on clue what they're up to—my sense is they're playing fast and loose with licensing terms, and they're going to start pissing people off… they already claimed they would "protect people from the viral nature of the GPL," which obviously didnt go down well with some in the Linux community…

On Nov 6, 2007, at 12:20 PM, John Fowler wrote:

Yuck.

btw Android does have Java in the stack. I'm assuming on the license part that not all is under the same license. Unless they have really big peaches.

Trial Exhibit 3211

UNITED STATES

SECURITIES AND EXCHANGE COMMISSION

Washington, D.C. 20549

FORM 10-K

(Mark One)

☒ **ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934**

For the fiscal year ended December 31, 2004

OR

☐ **TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934**

Commission file number: 000-50726

Google Inc.

(Exact name of registrant as specified in its charter)

Delaware	**77-0493581**
(State or other jurisdiction of incorporation or organization)	**(I.R.S. Employer Identification Number)**

1600 Amphitheatre Parkway
Mountain View, CA 94043
(Address of principal executive offices)
(650) 623-4000
(Registrant's telephone number, including area code)

Securities registered pursuant to Section 12(b) of the Act:

None

Securities registered pursuant to Section 12(g) of the Act:

Class A Common Stock, $0.001 par value
Class B Common Stock, $0.001 par value
(Title of class)

Indicate by check mark whether the registrant (1) has filed all reports required to be filed by Section 13 or 15(d) of the Securities Exchange Act of 1934 during The preceding 12 months (or for such shorter period that the registrant was required to file such reports), and (2) has been subject to such filing requirements for the past 90 days Yes [x] No [ ]

Indicate by check mark disclosure of Delinquent filers Pursuant to Item 405 of Regulation S-K is not contained herein, and will not be contained, to the best of the registrant's knowledge, definitive proxy or information statements incorporated by reference in Part III of this Form 10-K or any amendment to this Form 10-K [ ]

Indicate by check mark whether the Registrant is an accelerated filer (as defined in Rule 12b-2 of the Act) Yes [ ] No [x]

At December 31, 2004, the last business day of the Registrant's most recently completed fiscal quarter, there were 95,542,010 shares of Registrant's Class A common stock and 178,980,030 shares of Registrant's Class B common stock outstanding, and the aggregate market value of such shares held by non-affiliates of the Registrant (based upon the closing sale price of such shares on The Nasdaq National Market on December 31, 2004) was approximately $27,286,463,824. Shares of Registrant's Class A common stock and Class B common stock held by each executive officer and director and by each entity or person that, to the Registrant's knowledge, owned 5% or more of Registrant's outstanding common stock as of December 31, 2004 have been excluded in that such persons may be deemed to be affiliates of the Registrant. This determination of affiliate status is not necessarily a conclusive determination for other purposes.

At March 28, 2005, there were 114,754,458 shares of Registrants Class A common stock outstanding and 162,594,769 shares of Registrant's Class B common stock outstanding.

DOCUMENTS INCORPORATED BY REFERENCE

Portions of the registrant's Proxy Statement for the 2005 Annual Meeting of Stockholders are incorporated herein by reference in Part III of this Annual Report on Form 10-K to the extent stated herein.

* * *

company, we may find our recruiting efforts more challenging. The incentives to attract, retain and

motivate employees provided by our option grants may not be as effective as in the past and our current and future compensation arrangements, which include cash bonuses, may not be successful in attracting new employees and retaining and motivating our existing employees. In addition, we have recently introduced new stock award programs, and under these new programs new employees will be issued a portion of their stock awards in the form of restricted stock units. These restricted stock units will vest based on individual performance, as well as the exercise price of their stock options as compared to that of other employees who started at about the same time. These new stock awards programs may not provide adequate incentives to attract, retain and motivate outstanding performers. If we do not succeed in attracting excellent personnel or retaining or motivating existing personnel, we may be unable to grow effectively

Our CEO and our two founders run the business and affairs of the company collectively, which may harm their ability to manage effectively.

Eric, our CEO, and Larry and Sergey, our founders and presidents, currently provide leadership to the company as team. Our bylaws provide that our CEO and our presidents, will together have general supervision, direction and control of the company, subject to the control of our board of directors. As result, Eric, Larry, and Sergey tend to operate the company collectively and to consult extensively with each other before significant decisions are made. This may slow the decision-making process and a disagreement among these individuals could prevent key strategic decisions from being made in timely manner. In the event

our CEO and our two Founders are unable to continue to work well together in providing cohesive leadership, our business could be harmed.

We have a short operating history and a relatively new business model in an emerging and rapidly evolving market. This makes it difficult to evaluate our future prospects and may increase the risk that we will not continue to be successful.

We first derived revenue from our online search business in 1999 and from our advertising services in 2000, and we have only short operating history with our cost-per-click advertising model, which we launched in 2002. As result, we have very little operating history for you to evaluate in assessing our future prospects. Also, we derive nearly all of our revenues from online advertising, which is an immature industry that has undergone rapid and dramatic changes in its short history. You must consider our business and prospects in light of the risks and difficulties we will encounter as an early-stage company in new and rapidly evolving market. We may not be able to successfully address these risks and difficulties, which could materially harm our business and operating results.

We may have difficulty scaling and adapting our existing architecture to accommodate increased traffic and technology advances or changing business requirements, which could lead to the loss of users, advertisers and Google Network members, and cause us to incur expenses to make architectural changes.

To be successful, our network infrastructure has to perform well and be reliable. The greater the user traffic and the greater the complexity of our products and services, the more computing power we will need. In 2005, we expect to spend substantial amounts to purchase or lease data centers and equipment and to upgrade our technology and network infrastructure to handle increased traffic on our websites and to roll out new products and services. This expansion is going to be expensive and complex and could result in inefficiencies or operational failures. If we do not implement this expansion successfully, or if we experience inefficiencies and operational failures during the implementation, the quality of our products and services and our users experience could decline. This could damage our reputation and lead us to lose current and potential users, advertisers and Google Network members. The costs associated with these adjustments to our architecture could harm our operating results Cost increases loss of traffic or failure to accommodate new technologies or changing business requirements could harm our operating results and financial condition.

We rely on bandwidth providers, data centers or other third parties for key aspects of the process of providing products and services to our users, and any failure or interruption in the services and products provided by these third parties could harm our ability to operate out business and damage our reputation.

We rely on third-party vendors, including data center and bandwidth providers. Any disruption in the network access or co-location services provided by these third-party providers or any failure of these

third-party providers to handle current or higher volumes of use could significantly harm our business. Any financial or other difficulties our providers face may have negative effects on our business, the nature and extent of which we cannot predict. We exercise little control over these third- party vendors, which increases our vulnerability to problems with the services they provide. We license technology and related databases from third parties to facilitate aspects of our data center and connectivity operations including, among others, Internet traffic management services We have experienced and expect to continue to experience interruptions and delays in service and availability for such elements. Any errors, failures, interruptions or delays experienced in connection with these third-party technologies and information services could negatively impact our relationship with users and adversely affect our brand and our business and could expose us to liabilities to third parties.

Our systems are also heavily reliant on the availability of electricity, which also comes from third-party providers. If we were to experience major power outage, we would have to rely on back-up generators. These back-up generators may not operate properly through major power outage
and their fuel supply could also be inadequate during major power outage. This could result in
disruption of our business.

Interruption or failure of our information technology and communications systems could impair our ability to effectively provide our products and services which could damage our reputation and harm our operating results.

Our provision of our products and services depends on the continuing operation of our information technology and communications systems. Any damage to or failure of our systems could result in interruptions in our service. Interruptions in our service could reduce our revenues and profits, and our brand could be damaged if people believe our system is unreliable. Our systems are vulnerable to damage or interruption from earthquakes, terrorist attacks, floods, fires, power loss, telecommunications failures, computer viruses, computer denial of service attacks or other attempts to harm our systems, and similar events. Some of our data centers are located in areas with high risk of major earthquakes. Our data centers are also subject to break-ins, sabotage and intentional acts of vandalism, and to potential disruptions if the operators of these facilities have financial difficulties. Some of our systems are not fully redundant, and our disaster recovery planning cannot account for all eventualities. The occurrence of natural disaster, a decision to close a facility we are using without adequate notice for financial reasons or other unanticipated problems at our data centers could result in lengthy interruptions in our service.

We have experienced system failures in the past and may in the future. For example, in November 2003 we failed to provide web search results for approximately 20% of our traffic for period of about 30 minutes. Any unscheduled interruption in our service puts burden on our entire organization and would result in an immediate loss of revenue. If we experience frequent or persistent system failures on our web sites, our reputation and brand could be permanently harmed. The steps we have taken to increase the

reliability and redundancy of our systems are expensive, reduce our operating margin and may not be successful in reducing the frequency or duration of unscheduled downtime.

More individuals are using non-PC devices to access the Internet and versions of our web search technology developed for these devices may not be widely adopted by users of these devices.

The number of people who access the Internet through devices other than personal computers, including mobile telephones, hand-held calendaring and email assistants, and television set-top devices, has increased dramatically in the past few years. The lower resolution, functionality and memory associated with alternative devices make the use of our products and services through such devices difficult. If we are unable to attract and retain substantial number of alternative device users to our web search services or if we are slow to develop products and technologies that are more compatible with non-PC communications devices, we will fail to capture significant share of an increasingly important portion of the market for online services.

Payments to certain of our Google Network members have exceeded the related fees we receive from our advertisers.

We have entered into, and may continue to enter into, minimum fee guarantee agreements with small number of Google Network members. In these agreements, we promise to make minimum payments to the Google Network member for pre-negotiated period of time, typically from three months to year or more.

It is difficult to forecast with certainty the fees that we will earn under our agreements, and sometimes the fees we earn fall short of the minimum guarantee payment amounts. Also, increasing competition for arrangements with websites that are potential Google Network members could result in our entering into more of these minimum fee guarantee agreements under which guaranteed payments exceed the fees we receive from advertisers whose ads we place on those Google Network member sites. In each period to date, the aggregate fees we have earned under these agreements have exceeded the aggregate amounts we have been obligated to pay these Google Network members. However, individual agreements have resulted in guaranteed minimum and other payments to certain Google Network members in excess of the related fees we receive from advertisers. We expect that some individual agreements will continue to result in guaranteed minimum and other payments to certain Google Network members in excess of the related fees we receive from advertisers, which will adversely affect our profitability. However, we expect that the aggregate fees we will earn under agreements with guaranteed minimum and other payments will exceed the aggregate amounts we will be obligated to pay these Google Network members.

To the extent our revenues are paid in foreign currencies, and currency exchange rates become unfavorable, we may lose some of the economic value of the revenues in U.S. dollar terms.

As we expand our international operations, more of our customers may pay us in foreign currencies Conducting business in currencies other than U.S. dollars subjects us to fluctuations in currency exchange rates.

If the currency exchange rates were to change unfavorably the value of net receivables we receive in foreign currencies and later convert to U.S. dollars after the unfavorable change would be diminished This could have negative impact on our reported operating results Hedging strategies such as forward contracts options and foreign exchange swaps related to transaction exposures that we have implemented or may implement to mitigate this risk may not eliminate our exposure to foreign exchange fluctuations. Additionally, hedging programs expose us to risks that could adversely affect our operating results including the following:

- We have limited experience in implementing or operating hedging programs. Hedging programs are inherently risky and we could lose money as a result of poor trades.
- We may be unable to hedge currency risk for some transactions because of a high level of uncertainty or the inability to reasonably estimate our foreign exchange exposures.
- We may be unable to acquire foreign exchange hedging instruments in some of the geographic areas where we do business, or, where these derivatives are available, we may not be able to acquire enough of them to fully offset our exposure.

We may have exposure to greater than anticipated tax liabilities.

We are subject to income taxes and non-income taxes in variety of jurisdictions and our tax structure is subject to review by both domestic and foreign

taxation authorities. The determination of our worldwide provision for income taxes and other tax liabilities requires significant judgment and in the ordinary course of our business, there are many transactions and calculations where the ultimate tax determination is uncertain. Although we believe our estimates are reasonable the ultimate tax outcome may differ from the amounts recorded in our financial statements and may materially affect our financial results in the period or periods for which such determination is made.

We rely on insurance to mitigate some risks and, to the extent the cost of insurance increases or we are unable or choose not to maintain sufficient insurance to mitigate the risks facing our business, our operating results may be diminished.

We contract for insurance to cover certain potential risks and liabilities. In the current environment, insurance companies are increasingly specific about what they will and will not insure. It is possible that we may not be able to get enough insurance to meet our needs, may have to pay very high prices for the coverage we do get or may not be able to acquire any insurance for certain types of business risk. In addition, we have in the past and may in the future choose not to obtain insurance for certain risks facing our business. This could leave us exposed to potential claims. If we were found liable for significant claim in the future, our operating results could be negatively impacted Also, to the extent the cost of maintaining insurance increases, our operating results will be negatively affected.

Acquisitions could result in operating difficulties, dilution and other harmful consequences.

We do not have great deal of experience acquiring companies and the companies we have acquired have been small. We have evaluated and expect to continue to evaluate wide array of potential strategic transactions From time to time we may engage in discussions regarding potential acquisitions Any of these transactions could be material to our financial condition and results of operations In addition the process of integrating an acquired company business or technology may create unforeseen operating difficulties and expenditures and is risky The areas where we may face risks include:

- The need to implement or remediate controls, procedures and policies appropriate for a larger public company at companies that prior to the acquisition lacked these controls, procedures and policies.
- Diversion of management time and focus from operating our business to acquisition integration challenges.
- Cultural challenges associated with integrating employees from the acquired company into our organization.
- Retaining employees from the business we acquire.
- The need to integrate each other's company's accounting, management information, human resource and other administrative system to permit effective management.

Foreign acquisitions involve unique risks in addition to those mentioned above, including those related to integration of operations across different cultures and languages, currency risks, and the particular economic, political and regulatory risks associated with specific countries Also, the anticipated benefit of many of our acquisitions may not materialize. Future acquisitions or dispositions could result in potentially dilutive issuances of our equity securities, the incurrence of debt, contingent liabilities or amortization expenses, or write-offs of goodwill, any of which could harm our financial condition. Future acquisitions may require us to obtain additional equity or debt financing, which may not be available on favorable terms or at all.

We occasionally become subject to commercial disputes that could harm our business by distracting our management from the operation of our business, by increasing our expenses and, if we do not prevail, by subjecting us to potential monetary damages and other remedies.

From time to time we are engaged in disputes regarding our commercial transactions. These disputes could result in monetary damages or other remedies that could adversely impact our financial position or operations. Even if we prevail in these disputes, they may distract our management from operating our business and the cost of defending these disputes would reduce our operating results.

* * *

Trial Exhibit 5046

Sent: 4/17/2008 12:30 PM
From: sa3ruby@gmail.com
To: members@apache.org
Cc:
Bcc:
Subject: Re: and what if we accepted the damn FOU restriction?

On Thu, Apr 17, 2008 at 3:08 PM, Stefano Mazzocchi <stefano@apache.org> wrote:

Sam Ruby wrote:

On Thu, Apr 17, 2008 at 12:50 PM, Stefano Mazzocchi <stefano@apache.org> wrote:

Here's an idea:

1) we keep fighting at the JCP level but

2) in sign of good effort, we accept the TCK for java 5 with the FOU restriction as Sun offered

3) we run it against Harmony and pass

4) we release the binary versions of harmony under the Apache License 2.0 with the NOTICEs changed to imply the FOU restriction, indicating loud and clear that such restrictions are Sun's imposition even if we think they are in violation of our JSPA agreement

5) we *do not* add such notice to the source code and we do not distribute it bundled.

- o -

Why is this useful?

The FOU restriction applies to the binary but not the source code. Sun could still sue us implying that our 'svn' is distributing sources that can be turned into binary that might not pass the TCK but their position would be incredibly weak and would backfire tremendously on their own free software projects.

Why this is problematic?

---

a) The Harmony binaries with such field of use would *not* qualify as open source software under the OSI-rules. We will have to say that out loud (and blame Sun for it).

b) The ASF would be releasing non-OSI-compatible software for the first time in its history.

Why is this interesting at all?

The point would be to show Sun that FOU restrictions don't work, are toxic to an open development environment (including the ones they are are trying to create themselves) and can be easily bypassed.

If we comply to the terms but it's as easy as "svn co harmony; ./configure; make; make install" to get binary that has no FOU restriction.

- o -

Yes, there is the problem that we look like we caved and Sun won, which would bring negative ripples in the JCP EC and reduce our ability to leverage their restlessness for JCP reform.

None of this addresses my particular sticking points, which I can summarize as the following:

Thanks, this helps. See my comments below though.

> 1) The ASF will not make statements regarding intellectual property other than what we expressly know about, and that we disclose in Notice files and our Apache License.

I was working under the assumption that we could ignore the trademarks (avoid stating that we are compatible), use the org.apache.java + classload trick to avoid the java.* namespace and pretend that we don't know of any IP that we infringe until explicitly mentioned.

But what I was missing is the fact that the copyright on the API is real and hard to ignore.

Simply by implementing a class with the same signature of another, in another namespace and simply by looking at available javadocs could be considered copyright infringement, even if the implementation is clean room.

So, we are, in fact, infringing on the spec lead copyright if we distribute something that has not passed the TCK and *we know that*.

> [REDACTED]

We can claim (and have) that our milestone releases are just for development purposes… but then people still run with them.

In addition to all this, the JCK licenses that Geir have been presented with impose additional requirements. Remember the battle over notification requirements for Geronimo? As Geir puts it, this go around he might as well be "Negotiating with myself" for all the ground that Sun has been willing to cede in this discussion.

> 2) The ASF will not tell our licensees that they need to test their product with any given TCK, even if weakly phrased as "it is strongly recommended...."
>
> 3) The ASF will neither acknowledge nor speculate on the possible consequences of any loss of certification that might arise as a result of substantive modifications to ASF-licensed software.
>
> I will also note that over the course of the past 18 or so months, Geir has explored every possible suggestion that has been made to him. In addition to the concerns that we would rightly have on your suggestion, there still would be the issues listed above to deal with.

These are very valid points.

And they very much constrain what notification requirements we will accept. And, to date, Geir has yet to be presented with a TCK contract that conforms to these requirements.

> Stefano
>
> - Sam Ruby

Trial Exhibit 5048

From: Dan Bornstein. Sent:11/28/2006 4:53 PM.
To: Ramy Dodin.
Cc:
Bcc:
Subject: Re: java open source.

> I forgot to ask you the other day, will Sun's recent announcement about open sourcing java happen soon enough to benefit Android?

It's not about timing so much as details. The licensing that Sun is using for both SE and ME are incompatible with Android's needs. I'm happy to talk further about it in person.

> Also there are some pics from my travails at rdodin.com if you're interested.

I am, thanks!

-dan

Trial Exhibit 5114

From: Andy Rubin Sent: 3/28/2007 10:22 AM
To: Steve Horowitz
Cc:
Bcc:
Subject: Re: Latest material for CMCC's VIP Sha visit Wed morning.

He got different input from you and I. Additionally, I have a presentation which I show.

I need you to focus 100% on execution of the handset. We risk failure if you get distracted. We'll talk later about how I can help you. We are beyond out of time.

On Mar 28, 2007, at 10:17 AM, Steve Horowitz wrote:

Sorry … Was just trying to help correct some of the big errors. Wasn't sure how much time you'd have to review before the 9am meeting. Apologies.

-----Original Message-----

From: Andy Rubin
To: Steve Horowitz
Sent: Wed Mar 28 10:05:47 2007
Subject: Re: Latest material for CMCC's VP Sha visit Wed morning

Steve,

Please let me handle the partnership presentations!

On Mar 28, 2007, at 6:28AM, Steve Horowitz wrote:

Andy -- some comments below:

- Slide 12 has a minor typo, should read "Ads are only shown to users …" (currently says "Ads are only show to users"

- For slide 41 I have enclosed a new deck that has a more updated picture of our current device and screenshot. The version you have is about 1 year old. Also, I would remove the "UI Markup tool enables …" line from this slide.

- On slide 42, I would suggest changing the last two lines to something more like this:

"Powerful Java Application framework and optimized graphics system built on top of Linux kernel with the ability to run midlet-based content"

- On slide 44, remove the line about Skelmir (we have taken a different path for Java libraries)

- On slide 45, change line to "The Java platform will have the ability to run midlet based content" (remove CDC based). Also, change last line to "Java Application framework …."

- You can remove handset OS architecture slide in the appendix and this will be covered in the Android deep dive this diagram is old

Thanks,

-Steve

On 3/28/07, Andy Tian <andyt@google.com> wrote:

Hi all,

After reviewing the PPT with CMCC folks tonight, I made a few adjustments. Pls see attached for the latest PPT for tomorrow.

Just got notified that Alan won't arrive until 9:30am, so I will try to get the CMCC delegation to arrive at the conference room around 9:15, so we can spend some time to chat before Alan arrives and starts the

official meeting. This way we can do an official intro of Alan to CMCC VP Sha.

Thanks,

Andy

--

Andy Tian ???
Strategic Partnership Development ????
Google Inc.

Phone: +86.10.6250.3000
Direct: +86.10.6250.3846
Mobile: +86.1391.061.9810
Fax: +86.10.6250 3001
andyt@google.com

???????????1??,
?????6??,????, 100084

Tsinghua Science Park Building 6
No. 1 Zhongguancun East Road
Haidian District Beijing P.R.China. 100084

---

* * *

Trial Exhibit 5121

Google Web APIs (beta)

Terms and Conditions for Google Web API Service

Thank you for using the Google Web APIs™ service. By using this service ("Google Web APIs") you agree to be bound by the following terms and conditions (the "Terms and Conditions").

Personal and legitimate uses only

The Google Web APIs service is made available to you for your personal, non-commercial use only (at home or at work). You may only create a single account and must provide accurate identification, contact, and other information required as part of the registration process. You may not create any script or other automated tool that attempts to create multiple Google Web APIs accounts. And you may not use the search results provided by the Google Web APIs service with an existing product or service that competes with products or services offered by Google.

If you are interested in doing anything different than the foregoing, you must first obtain Google's written consent. If you fail to do so, Google reserves the right to take legal action.

Furthermore, you may not use Google Web APIs in any manner that either directly or indirectly violates any laws or proprietary rights. This includes laws and proprietary rights in the United States as well as in other countries.

If you have questions on your contemplated use or if you have comments on Google Web APIs or ideas on how to improve it, please email api-

support@google.com. Please note that by doing so, you also grant Google permission to use and incorporate your ideas or comments into Google Web APIs without further compensation.

Intellectual property

You agree not to remove, obscure, or alter Google's copyright notice, trademarks, or other proprietary rights notices affixed to or contained within Google Web APIs. You also acknowledge that Google owns all right, title and interest in and to Google Web APIs, including without limitation all intellectual property rights (the "Google Rights"). The Google Rights include rights to the following: (1) the APIs developed and provided by Google, (2) all software associated with the Google Web APIs server, and (3) the search results and spell checking you obtain when you use Google Web APIs. The Google Rights do not include the following: (1) third-party components used as part of Google Web APIs; or (2) software developed by you in conjunction with using Google Web APIs.

Publicity

So long as you comply with your obligations under this Agreement, you may indicate that a product or service that you created either used or is based on Google Web APIs provided that those products or services do not in Google's reasonable opinion (1) tarnish, infringe, or dilute Google's trademarks, (2) violate any applicable law, and (3) infringe any third-party rights. If you wish to use the GOOGLE trademark and/or logo in any other manner, you must first obtain Google's written consent.

Disclaimer of warranties

The Google Web APIs service is currently in beta form and has not been fully tested or debugged. Accordingly, Google disclaims any responsibility for any harm resulting from your use of Google Web APIs.

The Google Web APIs service is provided "as is," with no warranties whatsoever. Google expressly disclaims to the fullest extent permitted by law all express, implied, and statutory warranties, including, without limitation, the warranties of merchantability, fitness for a particular purpose, and non-infringement of proprietary rights. Google disclaims any warranties regarding the security, reliability, timeliness, availability, and performance of Google Web APIs.

You understand and agree that you use Google Web APIs at your own discretion and risk and that you will be solely responsible for any damages to your computer system or loss of data that results from the download or use of Google Web APIs.

Some states or other jurisdictions do not allow the exclusion of implied warranties, so the above exclusions may not apply to you. You may also have other rights that vary from state to state and jurisdiction to jurisdiction.

Limitation of liability

The Google Web APIs service is being provided free of charge. Accordingly you agree that Google shall have no liability arising from or based on your use of Google Web APIs.

Under no circumstances shall Google be liable to any user on account of that user's use or misuse of Google Web APIs. Such limitation of liability shall apply to

prevent recovery of direct, indirect, incidental, consequential, special, exemplary, and punitive damages whether such claim is based on warranty, contract, tort (including negligence), or otherwise, even if Google has been advised of the possibility of such damages). Such limitation of liability shall apply whether the damages arise from use or misuse of and reliance on the Google Web APIs, from inability to use Google Web APIs, or from the interruption, suspension, or termination of Google Web APIs (including such damages incurred by third parties). Such limitation shall apply notwithstanding a failure of essential purpose of any limited remedy and to the fullest extent permitted by law.

Some states or other jurisdictions do not allow the exclusion or limitation of liability for incidental or consequential damages, so the above limitations and exclusions may not apply to you.

Indemnification

You hereby agree to indemnify, defend and hold Google, and its officers, directors, agents, licensors and licensees (collectively, the "Indemnified Parties") harmless from and against any and all liability and costs incurred by the Indemnified Parties in connection with any claim arising out of your use of Google Web APIs, including, without limitation, reasonable attorneys' fees. You shall cooperate as fully as reasonably required in the defense of any claim. Google reserves the right, at its own expense, to assume the exclusive defense and control of any matter subject to indemnification by you.

Google's search service

The Google Web APIs service is designed to be used in conjunction with Google's search services. Accordingly, your use of Google Web APIs is also defined by Google's Terms of Service and Privacy Policy. In the event of a conflict between these Terms and Conditions and Google's Terms of Service, these Terms and Conditions shall prevail.

Term and termination

If you wish to terminate this Agreement, you may simply cease using the Google Web APIs service. In such a case, you must delete any search result information that you may have obtained from the Google Web APIs service. Google may terminate this Agreement (and your account) at any time, with or without cause.

General provisions

These Terms and Conditions will be governed by and construed in accordance with the laws of the State of California, without giving effect to the conflict of laws or provisions of California or your actual state or country of residence. Any dispute arising from these Terms and Conditions shall be adjudicated in the federal or state courts located in Santa Clara County, California. If for any reason a court of competent jurisdiction finds any provision or portion of these Terms and Conditions to be unenforceable, the remainder of these Terms and Conditions will continue in full force and effect. These Terms and Conditions constitute the entire agreement between the parties with respect to the subject matter hereof and supersede and replace all prior or contemporaneous understandings or agreements, written or oral, regarding such subject matter. Any waiver of any provision of

these Terms and Conditions will be effective only if in writing and signed by Google.

Trial Exhibit 5250

AdWords API beta

Terms & Conditions

This is a legal agreement between you and Google. By accepting these terms, you are representing that you have the authority to bind the party being issued a Developer Token for this AdWords API (you and that party collectively referred to as "you"). In exchange for use of and access to the proprietary AdWords API and its specifications you agree to be bound by the terms of these AdWords API terms and conditions (the "**AdWords API Agreement**"). The AdWords APIs are a feature of the Google AdWords program and any account management using the AdWords API is also governed by the AdWords terms and conditions between you and Google (your "**AdWords Terms**"). "Google" in this Agreement means the Google entity with which you have entered into your AdWords Terms (either Google Inc. or Google Ireland Limited) and its affiliates.

The AdWords API and the AdWords Specifications are, as applicable, the intellectual property and proprietary information of Google. Your right to use, copy and to retain your copy of the AdWords API and the AdWords API Specifications is contingent on your full compliance with this AdWords API Agreement. If you violate all or part of this AdWords API Agreement, your access to the AdWords API may be suspended or terminated without notice. If you decide to terminate your agreement to all or part of this AdWords API Agreement, you must cease all use of the AdWords API and destroy any copies of the AdWords API

Specifications, and if requested by Google, certify to Google such destruction.

I. Definitions

Using the "**AdWords API**" means: (A) the use of the mark-up language described in the AdWords API Specifications to (i) access Google servers through the AdWords API, (ii) send information to AdWords accounts using an AdWords API Client, or (iii) receive information from Google in response to AdWords API calls; and/or (B) distributing or developing an AdWords API Client.

For the purpose of this AdWords API Agreement, an "**account owner**" means the owner of record of an account or a party who the owner of record has expressly authorized to access and manage that account.

"**AdWords API Client**" means any software that can access or communicate with Google's servers using the AdWords API Specifications.

"**AdWords API Data**" means any data or content obtained from Google using the AdWords API.

"**AdWords API Specifications**" means all information and documentation Google provides specifying or concerning the AdWords API specifications and protocols and any Google-supplied implementations or methods of use of the AdWords API.

"**Internal-Use Only AdWords API Client**" means a Custom AdWords API Client developed only for one party who will be the sole user and owner of the client (the "**Developer**"). Internal-Use Only AdWords API Clients may not be

distributed and will cease to be an Internal-Use Only AdWords API Client if any or all rights in the client are transferred to, or all or a part of the client is used, by a party other than the Developer (except during development and testing in a non-live environment) or an agent of the Developer acting on Developer's behalf and using the client solely for the Developer. "**Custom**" for the purpose of this definition means that all copyright rights (including license rights) in the source and object code of the Internal-Use Only AdWords API Client (except for standard software APIs, modules, functions and libraries useful for creating or running software applications not related to Google) are owned solely by the Developer.

"**Third Party**" means a party other than Google or you (including without limitation any database, software or service owned by or under the control of a party other than Google or you).

II. AdWords API Use

1) Permission to Use. You may use the AdWords API to access Google only in accordance with the terms and conditions of this AdWords API Agreement and only with your own Developer Token.

2) Non-Compliant AdWords API Clients. You shall not use your Developer Token to access the AdWords API using an AdWords API Client that violates this AdWords API Agreement.

3) AdWords API Data.

a) Transfer of AdWords API Data. You shall not sell, redistribute, sublicense or otherwise disclose or transfer to any Third Party all or

any portion of the AdWords API Data (except that you may disclose the AdWords API Data from a particular account to the owner of that account if you are not the owner). The above sentence does not restrict the automated use or disclosure of AdWords API Data by an AdWords API Client that is in full compliance with this AdWords API Agreement.

b) API Materials. You shall not disclose all or part of any AdWords API Specification or your Developer Token to any Third Party except your agents using such information solely on your behalf, solely in accordance with this AdWords API Agreement and under a written duty of confidentiality no less strict than this AdWords API Agreement. You take responsibility for any and all use and disclosure of AdWords API Specifications and your Developer Token obtained through you.

c) Security. You shall use all reasonable efforts to keep all AdWords API Data, your Developer Token and AdWords API Specifications in a secure environment at all times according to the highest security standards. All data transfer using the AdWords API must be secured using at least 128 Bit SSL encryption.

d) Data Collection. You shall not use any automated means other than the AdWords API (for example scraping and robots) to access, query or otherwise collect Google-related information from Google, the AdWords Program or any website owned or operated by Google or a Google

partner site that displays Google advertising (collectively "Google Scraping").

4) No Guaranteed Access. Google may suspend or terminate your access to the AdWords API for any or no reason and will bear no liability for such decision. It is solely your responsibility at all times to backup your data and to be prepared to conduct your account without access to the AdWords API. GOOGLE DOES NOT REPRESENT OR WARRANT, AND SPECIFICALLY DISCLAIMS, THAT THE ADWORDS API WILL BE AVAILABLE WITHOUT INTERRUPTION.

5) Google Monitoring. You acknowledge that Google may monitor any AdWords API activity for the purpose of ensuring quality, improving Google products and services and compliance with these terms. You shall not try to interfere with such monitoring or otherwise obscure from Google your AdWords API activity. Google may use any technical means to overcome such interference.

III. AdWords API Client Development and Distribution

1) Permission to Use. You may use the AdWords API and AdWords API Specifications to develop and distribute an AdWords API Client only in accordance with the terms and conditions of this AdWords API Agreement.

2) Non-Compliant AdWords API Clients. Any AdWords API Client (and its development and distribution) must comply with this AdWords API Agreement.

a) Developer Token Domain.

i) Subject to the exceptions in the below subsections (ii)-(iv), an AdWords API Client must not enable or allow any party to access or use: (a) the account, passwords, AdWords API Data or Developer Tokens or any other information or another party; (b) any data obtained through Google Scraping; or (c) any AdWords API Specifications.

ii) AdWords API Clients may disclose AdWords API Data of an account to the account owner.

iii) AdWords API Clients may use for the then-current user (but not disclose in raw form) AdWords API Data collected using the then-current user's Developer Token.

iv) AdWords API Clients may transfer the AdWords API Data they collect to a database exclusively accessible, owned and controlled by the party who owns the Developer Token with which that data was obtained. Such databases may only be accessed by AdWords API Clients in conformance with this agreement or intermediary tools whose sole purpose is to analyze or supply data to an AdWords API Client in conformance with this AdWords API Agreement. Such databases must be fully secure at all times.

b) Aggregate Information. Notwithstanding anything to the contrary in this AdWords API Agreement, AdWords API Clients may only use or transfer AdWords API Data as expressly allowed in Section III of this API Agreement and, except in the case of a disclosure of account information to that account owner, only if all the data is in

aggregate form that cannot be correlated to specific users or accounts.

c) Personally Identifiable Information. The AdWords API Client must not collect personally identifiable information of any party unless it first informs the user about the types of information being collected and how that information may be used and then obtains the user's express permission.

d) Co-Mingling. The AdWords API Client must not co-mingle or associate any AdWords API Data or AdWords API input fields with the content of third parties. All AdWords API Data must be displayed in the AdWords API Client so that it is visually separate from any non-Google content. This Section III(2)(d) does not apply to Internal-Use Only AdWords API Clients.

e) Delayed Data. The AdWords API Client shall prominently disclose in the AdWords API Client the extent to which any displayed information is on delay.

f) Most Current API. All AdWords API Clients must only use a version of the AdWords API that was the most current version of the AdWords API within the 3 months preceding use (a "**Current AdWords API Version**"). Any less-current AdWords API Clients must be updated and must not be used, distributed, supported or maintained.

g) Full Functionality. All AdWords API Clients must expose at least as much campaign management functionality as is available in the

AdWords API and the standard AdWords Web-based user interface of any version of AdWords during the 3 months preceding use for the functions being affected by the AdWords API Client (which must include, without limitation, giving the end-user the ability to make calls to all parameters made available by the AdWords API Specifications for that particular functionality). For example, if a particular AdWords API Client enables bid-management, all aspects of AdWords bid management and all API calls related to AdWords bidding must be enabled by that client. This Section III(2)(g) does not apply to Internal-Use Only AdWords API Clients.

h) Security. Each AdWords API Client must have adequate protections in order to keep secure and prevent the interception of all AdWords API Data, Developer Tokens and AdWords API Specifications. All such information must be kept in a secure environment at all times according to the highest security standards. All AdWords API Clients must transmit data with a protocol at least as secure as 128 Bit SSL encryption.

i) Data Collection. The AdWords API Client must not enable Google Scraping.

j) Compliance with Law. The AdWords API Client must comply with all applicable government laws, rules and regulations and any third party's rights and must not operate in a manner that is, or that a user of the AdWords API Client would reasonably consider, deceptive, unethical, false or misleading.

k) Duty not to Interfere. The AdWords API Client must not interfere or attempt to interfere in any manner with the proper working of the AdWords API. Each AdWords API Client must pass a consistent and accurate identification of itself to Google in the "user-agent" header or similar field as outlined in the AdWords API Specification.

l) Google Monitoring. The AdWords API Client must not, and must not attempt to, interfere with Google monitoring of AdWords API activity or otherwise obscure from Google AdWords API activity. Google may use any technical means to overcome such interference, including without limitation suspending or terminating access of the AdWords API Client.

3) API Client Branding.

a) Logo Requirement. Each screen of an AdWords API Client that displays AdWords API Data or otherwise facilitates account management through the AdWords API must somewhere above the fold on that screen indicate that such management or data is for Google AdWords and such screen must display the logo available at https://adwords.google.com/select/images/google_small.gif (the "**AdWords Logo**") but only in compliance with this AdWords API Agreement and the FAQ on API Logo use at https://adwords.google.com/select/images/google_small.gif. This Section III(3)(a) is not required for Internal-Use Only AdWords API Clients

b) Trademark License. Subject to the terms and conditions of this AdWords API

Agreement Google grants you a non-exclusive worldwide license to display the AdWords Logo as set forth in Section III(3)(a) and only in an AdWords API Client that complies with the terms of this AdWords API Agreement.

c) Logo Co-mingling. You shall not display an AdWords Logo on any screen that contains: (i) a logo or trademark of a Third Party, or (ii) data or information of a Third Party (other than the then-current user of the AdWords API Client) or otherwise not related to the Google AdWords API. Section III(3)(c)(ii) does not apply to Internal-Use Only AdWords API Clients, but such clients may not display non-Google data in a manner that would lead a person to reasonably attribute such data to Google.

d) Brand Feature Rules. Except as expressly set forth above, this AdWords API Agreement does not grant you any rights to Google Brand Features (defined below). Google may withdraw any license to any Google Brand Features at any time for any or no reason. You shall not alter Brand Features in any way at any time (for example, changing color or size) without Google's prior written permission. You shall not display Brand Features on, or associate Brand Features with, any adult or illegal content. Goodwill in the Brand Features will inure only to Google's benefit and you obtain no rights with respect to any of them. You irrevocably assign and must assign to Google any right, title and interest that you obtain in any of Google's Brand Features. You must not at any time challenge or assist others to challenge Brand Features or their

registration (except to the extent you can't give up that right by law) or attempt to register any trademarks, marks or trade names confusingly similar to Google's. "**Brand Features**" means the trade names, trademarks, service marks, logos, domain names and other distinctive brand features of Google, including without limitation the AdWords Logos. This paragraph survives any termination or expiration of this agreement.

IV. The AdWords APIs in General

1) No Continuous Standard. Google may suspend or terminate your access to the AdWords API, or change any or all of the AdWords API, protocols or methods of access to the AdWords API or the AdWords API Specifications for any or no reason and is not liable to you for such decision.

2) Client Software Principles. Your development and distribution of any software application that accesses Google, including without limitation your development or distribution of any AdWords API Client, must at all times comply with the software principals set forth at http://www.google.com/corporate/software_principles.html.

3) Compliance with Law and Policies. You are responsible for complying with all applicable government laws, rules and regulations, all third party rights and all Google policies. You shall not use the AdWords API in a manner that violates such laws, rules and regulations, third parties' rights or any Google policies or in a manner that is deceptive, unethical, false or misleading.

4) Duty not to Interfere. You shall not interfere or attempt to interfere in any manner with the proper working of the AdWords API.

5) Usage and Quotas. Google may, in its sole discretion, set a quota of operations on your AdWords API usage based on, among other things, the AdWords spend history of the accounts under management by your Developer Token. You shall not attempt to exceed automated use-quota restrictions. You shall not attempt to aggregate your use or representation of AdWords accounts primarily for the purpose of manipulating your AdWords API use-quotas.

6) Billing. Google may, in the future and in its sole discretion, charge fees for types or quantities of AdWords API services. Service fee rates will be posted in the AdWords API FAQ at http://www.google.com/support/adwordsapi, and may change in Google's sole discretion at any time. You are responsible for AdWords API service fees based on service fee rates for which you have had at least 30 days notice (acceptable notice for fee changes includes without limitation posting a notice to your AdWords account or emailing the address of your AdWords or Developer Token account). You shall pay any fees and charges for Google's provision of AdWords API services as described in the AdWords API FAQ at http://www.google.com/support/adwordsapi on the same terms set forth in your AdWords Terms for the account under which you acquired your Developer Token. Charges are exclusive of taxes and you are responsible for any taxes or government charges. All AdWords API charges will

appear in the invoice for the account under which you acquired your Developer Token on a single aggregated line item. Notwithstanding anything to the contrary in your AdWords Terms, this charge is to you, not your customers, and you are fully responsible for any such charges you accrue whether you have collected fees from customers or not.

7) No Implied Rights. Other than expressly granted herein, this AdWords API Agreement does not grant either party any intellectual property or other propriety rights. You hereby release and covenant not to sue Google and its corporate affiliates and any of their licensees, assigns or successors, for any and all damages, liabilities, causes of action, judgments, and claims (a) pertaining to any intellectual property you develop that is based on, uses, or relates to the AdWords API; and (b) which otherwise may arise in connection with your use of, reliance on, or reference to the AdWords API. As between you and Google, Google and its applicable licensors retain all intellectual property rights (including all patent, trademark, copyright, and other proprietary rights) in and to the AdWords API Specifications, all Google websites and all Google services and any derivative works created thereof. All license rights granted herein are not sublicenseable, transferable or assignable unless otherwise stated herein.

8) Non-exclusive. This AdWords API Agreement is a non-exclusive agreement. You acknowledge that Google may be developing and may develop products or services that may compete with this

AdWords API, AdWords API Clients or any other products or services.

9) Third Party Opt Out. You must, at all times, provide to any customers for whose accounts you are accessing through the AdWords API the ability to easily and quickly (no longer than 3 business days after customer notice to you) disassociate their AdWords campaigns from your services and Developer Tokens and regain exclusive control of their AdWords accounts.

10) Indemnification. You shall indemnify, defend and hold Google, its agents, affiliates, and licensors harmless from any claim, costs, losses, damages, liabilities, judgments and expenses (including reasonable fees of attorneys and other professionals), arising out of or in connection with any claim, action or proceeding (any and all of which are "**Claims**") arising out of or related to any act or omission by you in using the Google AdWords API, or relating to the development, operation, maintenance, use and contents of an AdWords API Client, including by not limited to any infringement of any third-party proprietary rights. At Google's option, you shall assume control of the defense and settlement of any Claim subject to indemnification by you (provided that, in such event, Google may at any time thereafter elect to take over control of the defense and settlement of any such Claim, and in any event, you shall not settle any such Claim without Google's prior written consent).

11) Termination. Any licenses contained in this AdWords API Agreement will terminate

automatically without notice if you fail to comply with any provision of this AdWords API Agreement. Google reserves the right to terminate this Agreement or discontinue the AdWords API or any portion or feature thereof for any or no reason and at any time without liability to you. Upon any termination or notice of any discontinuance, you must immediately stop and thereafter desist from using the AdWords API or distributing or developing AdWords API Clients and delete all AdWords API Specifications in your possession or control (including without limitation from your AdWords API Client and your servers). The provisions of Sections I, II 2, 3 (b)-(d), 4 and 5, III 2, 3(a), (c) and (d) and IV will survive any termination of this AdWords API Agreement and will continue to bind you in accordance with their terms.

12) Modification. Google may modify any of the terms and conditions contained in this API Agreement, at any time and in its sole discretion, by posting a change notice to your account, changing the agreement linked from the AdWords FAQ, emailing the email address of your AdWords account or Developer Token account or otherwise notifying you. IF ANY MODIFICATION IS UNACCEPTABLE TO YOU, YOUR ONLY RECOURSE IS TO TERMINATE THIS AGREEMENT. YOUR CONTINUED USE OF THE ADWORDS API, CONTINUED POSSESSION OF A COPY OF THE ADWORDS API SPECIFICATIONS OR CONTINUED DEVELOPMENT OR DISTRIBUTION OF AN ADWORDS API CLIENT FOLLOWING POSTING OF A CHANGE NOTICE OR NEW AGREEMENT ON OUR SITE

WILL CONSTITUTE BINDING ACCEPTANCE OF THE CHANGE.

13) Disclaimer and Limitation of Liability. GOOGLE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION FOR NONINFRINGEMENT, MERCHANTABILITY AND FITNESS FOR ANY PURPOSE. GOOGLE SHALL HAVE NO DIRECT, CONSEQUENTIAL, SPECIAL, INDIRECT, EXEMPLARY, PUNITIVE, OR OTHER LIABILITY WHETHER IN CONTRACT, TORT OR ANY OTHER LEGAL THEORY, UNDER THIS ADWORDS API AGREEMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LIABILITY AND NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY. IN THE EVENT THAT THE ABOVE IS NOT ENFORCEABLE, GOOGLE'S AGGREGATE LIABILITY UNDER THIS AGREEMENT IS LIMITED TO AMOUNTS PAID OR PAYABLE TO GOOGLE BY YOU FOR THE ADWORDS API SERVICES IN THE MONTH PRECEDING THE CLAIM.

14) Miscellaneous. The AdWords API Agreement must be construed as if both parties jointly wrote it, governed by California law except for its conflicts of laws principles and adjudicated in Santa Clara County, California. The AdWords API Agreement constitutes the entire agreement between the parties with respect to the subject matter hereof. Any conflicting or additional terms contained in additional documents or oral discussion are void. You may grant approvals, permissions and consents to Google by email, but any

modifications by you to this AdWords API Agreement must be made in a writing (not including email) executed by both parties. Any notices to Google must be sent to our corporate headquarters address as set forth on our website via first class or air mail or overnight courier, and is deemed given upon receipt. A waiver of any default is not a waiver of any subsequent default. Unenforceable provisions will be modified to reflect the parties' intention, and remaining provisions of the AdWords API Agreement will remain in full effect. Customer may not assign any of its rights hereunder and any such attempt is void. You and are not legal partners or agents, but are independent contractors.

January 25th, 2005

Trial Exhibit 5322

From: Rich Miner Sent: 10/23/2006 01:13 PM
To: Andy Rubin
Cc:
Bcc:
Subject: Re: Android announce at 3GSM

Andy,

I think we probably should here are some random thoughts on why I think that along with a dump of issues:

Why announce:

- if we wait then other soln's will have a much better chance of getting traction. I don't think that they will be that much further along in terms of a complete platform (I am thinking QT, Access, etc). As a case in point, if we were not doing what we are doing, SavaJe would prob have gotten more funding.

- Assist with partnerships. The hardware guys (like Moto) will want to get their ducks in a row before we announce. They will realize that we only have so much bandwidth and if they don't jump on board then others will take their place. The software guys will all wan to be a part of it.

- Help with carrier deals. All will realize that the chance of doing something exclusive with us in their key markets will be near impossible once we launch. Having a launch date will be a strong sales tool to have contracts completed.

What do we announce: pick one or more of the following:

1. Google Phone (Sooner?

2. OS
3. Alliance
4. Carrier Partners (signed on to-date)
5. OEMs (would moto show FNDR that far before launch, I doubt it)
(think we should at least announce 1,2 & prob 3).

Issues:

- We will have to show a phone. That means that we will have significant preassure to shop sooner. We better make sure we want to ship it.
- It will be hard to have press photo's of the device and UI before the event.. given how much work we still have to do. I think this can be ok, but we have to manage it.
- We will need FT staff to handle the requests that come in.
- We will need our 3rd party developer program to be ready
- We will need to have accurate dates for us to ship units to developers and for FCS for the 1st devices
- Might we still do an Orkut phone under this scenario?
- do we show other designs (dream, etc. or just keep those for later)

Negatives:

- puts lots of pressure on schedule (prob as much a positive)
- gives a heads up to competition (but what are they going to do about it)
- we need to get the UI nailed down!

Anyway, just some initial random thoughts. I will keep them coming.

How about Chinese tomorrow night?

..Rich

On 10/23/06, Andy Rubin <arubin@google.com> wrote:

What do you think? Should we announce?

Begin forwarded message:

From: Jim Holden <jholden@google.com>
Date: October 23, 2006 6:01:14 AM PDT
To: "Christian Hernandez Gallardo" <christianh@google.com>
Cc: Andy Rubin <arubin@google.com>
Subject: Re: Fwd: Android announce at 3GSM

[adding Andy]

Christian,

I agree with you. There is no reason to hold back on an Android announcement at 3GSM if the Android team is ready.

Having said that, we will need to plan and execute the messaging very very carefully. It is important for the world to understand what Android is not trying to do (or at least the key players both operators and oems to understand that). We want folks to see how Android complements the market and can bring distinct benefits for operators and oems.

This is not easy messaging. The message is both subtle and forward-looking, and is also slightly different for operators and oems. It is far easier for the press to just write, "Google phone to take over the world," than to get the reality across.

If the Android team feel that they are ready to announce at the time of 3GSM, then we need to make sure that we have before that time had all the senior private conversations with our wireless partners and key potential partners, and then make a pre-announcement to them and private Q&A shortly before, so that they have it straight at least in their own heads.

Andy, working around your surgery of course, we would like to get you confirmed for a week in November so that we can come together for some top level Android discussions with O2 Germany, Vodafone, Carphone Warehouse, Brightstar, Telecom Italia and Telefonica.

Thanks,
Jim

At 13:44 23/10/2006, Christian Hernandez Gallardo wrote:

Jim, FYI on a dialogue with Andy over the weekend. I'd like to discuss with you to ensure that we are on the same page...

---------- Forwarded message ----------

From: Christian Hernandez Gallardo <christianh@google.com >
Date: Oct 21, 2006 8:14 PM
Subject: Re: Android announce at 3GSM
To: Andy Rubin < arubinggoogle.com >

Andy,

My opinion is that we should announce at 3GSM because:

- its the one week where the world press focuses on mobile, and if the objective is to shake up the industry, this is the event with the most impact.

- It enables us to have much easier and more open conversations with potential partners (operators, OEMs, retailers) to generate demand for Sooner

- it does not affect our existing business. If appropriately positioned our mobile search efforts are separate and distinct from Android. TIM, as an example, already understands the value we offer for mobile search and separately, the value a google phone-experience provide. Same with Motorola who will deliver an Android phone but a dozen of search-enabled phones. I believe Jim and Emmanuel Sauquet (APLA Biz Dev lead) would agree.

The question becomes, when do we announce if not at 3GSM? Not until CTIA?

Hope that adds one more voice to the discussion.

* * *

Trial Exhibit 5562

From: Hiroshi Lockheimer Sent: 3/27/2007 10:50 PM
To: Jim Hutchison; Kipping, David
Cc:
Bcc:
Subject: some materials from today.

Hi guys,

Thank you very much for a super productive meeting today. I am thinking we should meet in person again next week, in San Diego. How do Weds or Thurs sound?

I've attached the PowerPoint from today for your reference. Also, the who/what/when spreadsheet is also attached. More tomorrow, but wanted to get you these right away.

Hiroshi

Project Android
Qualcomm Meeting 3/27/07

Project Android

Open source mobile platform for smart phones

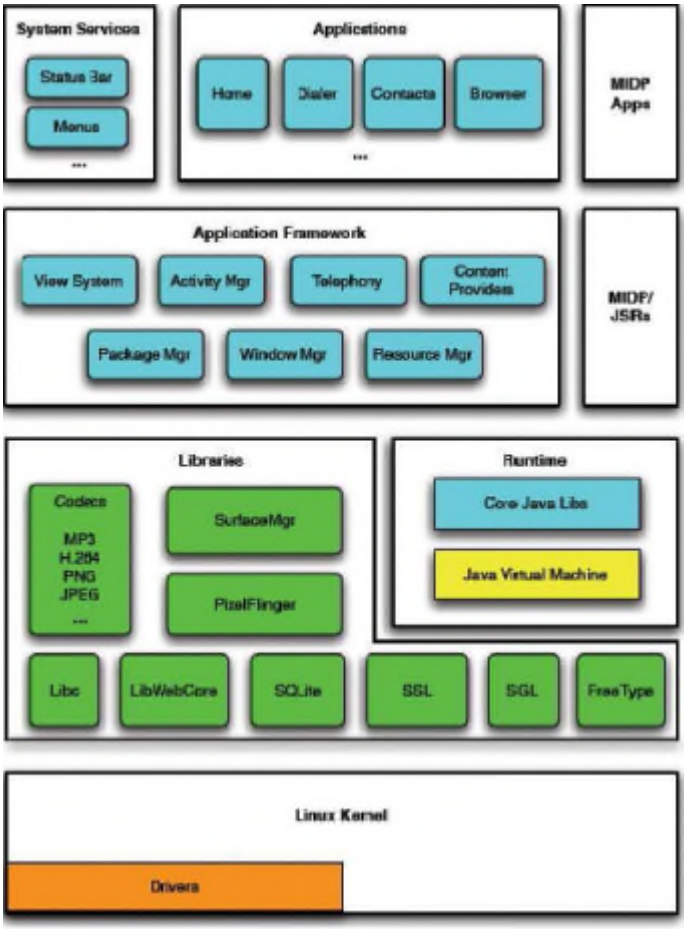
Linux 2.6 kernel and drivers

Java virtual machine for middleware and apps

Android Advantages

FIRST truly open, freely available Linux-based phone stack built from the ground up offering:

- Great phone experience
- Integrated Google applications
- Powerful, simple Java Application Framework
- Scalable, customizable applications and user experience
- Takes advantage of existing Linux driver model
- Advanced graphics system & rich media experience
- Advanced, standards-based Web Browser
- Complete phone solution



Kernel and Drivers

- Linux 2.6 kernel & custom boot loader
- Drivers
  - Display
  - Camera
  - Bluetooth
  - Wifi
  - USB
  - Keypad

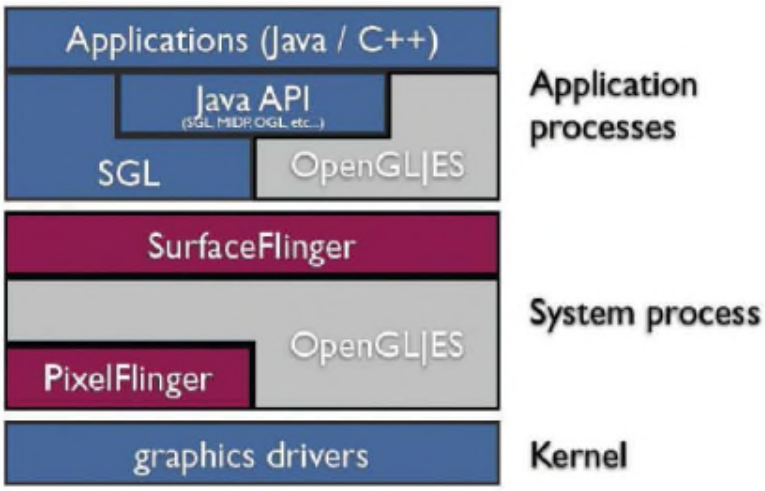
M-Systems
Audio
Binder (IPC)
Power Management

* * *

Graphics

- SGL Library for application drawing
- OpenGL | ES for gaming, 3D
- Surface Flinger for application sharing of frame buffer
- Pixel Flinger for low level rendering

Graphics Architecture



SGL Library

- SGL stands for Scalable Graphics Language
- Application 2D drawing and geometry API
- Has Vectors, Bitmaps, Text, Fonts, Gradients, Antialiasing

- Imaging model similar to PostScript

* * *

Trial Exhibit 5585

From: Steve Horowitz Sent: 7/7/2006 5:13 PM
To: Rich Miner
Cc:
Bcc:
Subject: Docs for LG

Activities and Intents

* * *

Project Android

- Open source mobile platform for smart phones
- Linux 2.6 kernel and drivers
- Java virtual machine for middleware and apps

Android Advantages

FIRST truly open, freely available Linux-based phone stack built from the ground up offering:

- Great phone experience
- Integrated Google applications
- Powerful, simple Java Application Framework
- Scalable, customizable applications and user experience
- Takes advantage of existing Linux driver model
- Advanced graphics system & rich media experience
- Advanced, standards-based Web Browser
- Complete phone solution

Overview

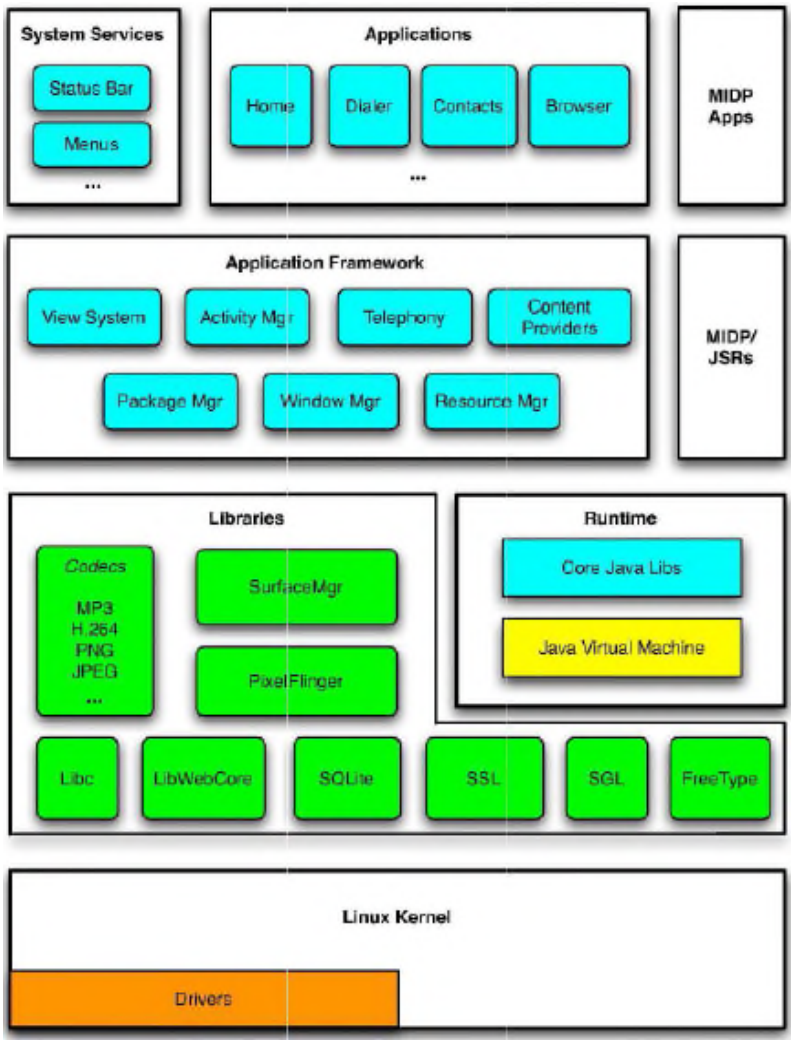
- Hardware requirements

- System Architecture
- Kernel and Drivers
- Processes
- Graphics, Audio & Video
- Telephony
- App Framework
- Applications & UI
- Developer Tools

* * *

Optional Hardware

- QWERTY keyboard
- Secondary display
- Bluetooth 2.0 EDR
- 802.1 lg
- 3D/Hardware Graphics acceleration
- GPS
- Baseband: WCDMA (HSDPA)



System Software

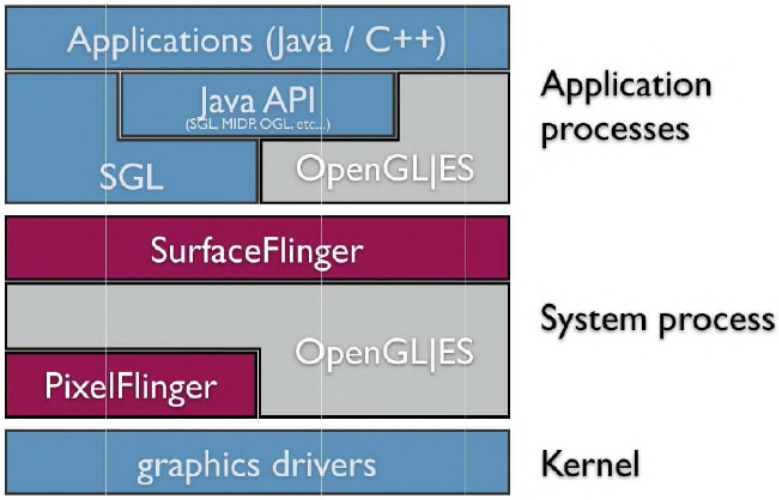
* * *

Graphics

- SGL Library for application drawing
- OpenGL | ES for gaming, 3D

- Surface Flinger for application sharing of frame buffer
- Pixel Flinger for low level rendering

Graphics Architecture



SGL Library

- SGL stands for Scalable Graphics Language
- Application 2D drawing and geometry API
- Has Vectors, Bitmaps, Text, Fonts, Gradients, Antialiasing
- Imaging model similar to PostScript

* * *

Trial Exhibit 5586

From: Rich Miner Sent: 9/11/2008 1:27 PM
To: Dan Zheng
Cc: Patrick Brady; ericchu@google.com; Lan Roche
Bcc:
Subject: Re: draft deck for AT&T meeting

latest…

On Wed, Sep 10, 2008 at 7:18 PM, Dan Zheng <danz@google.com> wrote:

I repurposed the deck from Patrick for tomorrow's meeting with AT&T. I moved all the non-relevant slides to Appendix. This is first cut, so please feel free to edit.

Dan

--
Dan Zheng
New Business Development
Google Inc.
650 253 4352

AT&T / Google Android Collaboration
September 2008

Agenda

- Review status of Android within AT&T
  - Open issues
  - Path to address those issues
- Quick product update & Overview
  - Demo
- Android Security overview
- Specific review of AT&T's questions

* * *

Android Security and Focus

Android's security model is as good and in some ways better than other commercially available smart phone platforms

Focus Areas

- Protect User Data
- Protect System Resources (Including network)
- Provide Application Isolation

Approach

- Robust security through Linux OS kernel
- Sandbox all applications
- Secured inter-process communication through a Binder facility
- Application-defined & user-granted permissions
- Application signing

Android Security: Process Partitioning

Android enforces security between applications and system at the Linux process level. Each application and service runs in a separate process*

- Each process gets its own framework, VM, libc, etc.
- Any compromise is isolated to a single process

Android Security: Data Partitioning

Android uses separate file-system partitions to protect core platform from 3rd party applications and data:

System Partition

- Contains all core system binaries (kernel, drives, native libraries, Android platform, VM, sqlite, etc)
- Contains all pre-loaded applications
- Mounted as read-only—contents cannot be modified or re-written

---

* Applications signed with the same key can choose to share the same process

Data Partition

- Contains all 3rd party applications
- Contains all applications and user data modified at runtime
- Each application gets its own data space. Files created in that space are not shared with other apps unless done so explicitly.

* * *

Trial Exhibit 6053

CNBC NEWS RELEASES

CNBC's Jim Cramer Interviews Google Inc. Chairman & CEO Eric Schmidt on "Mad Money with Jim Cramer" (Transcript Included)

Jennifer Dauble
Friday, 15 Aug 2008 | 9:40 AM ET

When: AUGUST 13TH AT 1:30PM ET
Where: CNBC'S "MAD MONEY W/ JIM CRAMER"

Following is the unofficial transcript of a CNBC interview with Google Inc. Chairman & CEO Eric Schmidt on "Mad Money w/Jim Cramer."

All references must be sourced to CNBC's "Mad Money w/Jim Cramer."

CRAMER: I'M CRAMER, COMING TO YOU LIVE, "MAD MONEY: AT THE HALF." THERE'S ONE STOCK I'VE CHAMPIONED MORE THAN ANY OTHER SINCE BECOMING A TV PERSONALITY THAT STOCK IS GOOGLE. I CALLED THIS A BUY FROM THE MOMENT IT CAME PUBLIC. I SAID IT WOULD BE A TRIPLE. I WAS COMPLETELY WRONG. IT WAS UP A SEVEN-FOLDER. MY RECORD ON GOOGLE WASN'T PERFECT BUT NOW WHEN I SAY IT'S TIME TO BUY THIS ONE AND YES, HAND OVER FIST. IT'S PROBABLY A GOOD IDEA TO LISTEN UP AND LISTEN GOOD. GOOGLE GOT CREAMED AFTER ITS SO-CALLED BAD QUARTER BACK ON JULY 17TH, I THOUGHT THE DECLINE WAS STUPID. I WOULD DECLARE IT TO BE AS STUPID AS BRICKS OR PLYWOOD. BECAUSE OF THESE

PANICERS, YOU CAN PICK IT UP FOR ABOUT 500 BUCKS A SHARE. THAT'S A STEAL. TECH IS NOW BACK IN FAVOR. THANKS TO A DECLINE OF OIL, AND THE TRIUMPH OF DEFLATION OVER INFLATION THIS WAS A STUPID SELL OFF FOR USAT FANS. THE BEARS READ THE QUARTER THE WAY THEY WANT TO. AND NOW THE STOCK IS TRADING AT 20 TIMES MY FORWARD EARNING ESTIMATES DESPITE TORRID 30% LONG-TERM GROWTH RATE. THE ISSUE IS, AM I TOO BULLISH? ARE THE COSTS OF TRAFFIC GOING UP? IS THERE MORE EARNING CYCLICALITY THAN I THOUGHT? HAS DOMESTIC HIT A WALL? IS THE MANAGEMENT TEAM NOT INVESTOR FRIENDLY ENOUGH. IS ANDROID THE BIG NEW PHONE FOR REAL? THERE ARE QUESTIONS WE NEED ANSWERED BEFORE I GIVE IT TWO THUMBS UP. THAT'S WHY I'M THRILLED TO BRING ON ERIC SCHMIDT. WE WERE NOT DOING THAT TWO MINUTE CEO DRILL THAT TV IS INFAMOUS FOR. THIS MAN HAS CHANGED THE WORLD. A WORLD THAT I BELIEVE IS DIVIDED INTO BG AND AG YOU FIGURE OUT WHAT THOSE MEAN. MR. ERIC SCHMIDT.WELCOME.

SCHMIDT: I THINK THIS IS YOURS.

CRAMER: YOU GOT THE TOYS.

SCHMIDT: YOUR OFFICE LOOKS LIKE THE ONES AT GOOGLE.

CRAMER: THEY'VE THE SPIRIT. THANK YOU. YOU HAVE A STOCK THAT'S AT 500 BUCKS. I GET STOPPED ALL THE TIME BY PEOPLE WHO KNOW I LOVE GOOGLE AND SAY, WILL YOU

GET THE GUY TO SPLIT THE STOCK. I CAN'T AFFORD 500 BUCKS WHY WON'T THEY SPLIT IT. TELL ME HOW STUPID THAT IS.

SCHMIDT: THIS IS NEW YORK, THEY CAN AFFORD $500.

CRAMER: I LIKE THAT BUT I ACTUALLY I GET THIS QUESTION WHEN I AM IN NORTH CAROLINA, TOO.

SCHMIDT: WE'RE NOT GOING TO SPLIT IT. PEOPLE THINK THE VALUE OF THE STOCK IS REALLY THE DOLLARS, SO WE KEEP IT HIGH.

CRAMER: I THINK THAT'S GOOD. I LIKE INDIVIDUALS TO BUY ONE SHARE. BERKSHIRE HATHAWAY DID THE SAME THING.

SCHMIDT: IT WORKS WELL.

CRAMER: PEOPLE FEEL YOU DON'T PROVIDE ENOUGH GUIDANCE, MANAGEMENT SHEPPARDING.

SCHMIDT: WE DON'T PROVIDE ANY GUIDANCE.

CRAMER: THAT'S BECAUSE—

SCHMIDT: WE DON'T WANT TO GET IN THE WAY OF RUNNING THE BUSINESS. IF WE STARTED GIVING QUARTERLY GUIDANCE THE COMPANY WOULD FOCUS ON THE QUARTER RATHER THAN TRYING TO CHANGE THE WORLD.

CRAMER: TOTALLY TRUE I WISH OTHER PEOPLE WOULD DO IT. THERE ARE 31 OUT OF 33 PEOPLE RECOMMENDING THE STOCK, SO IT

MAY NOT HAVE MATTERED YOU'RE NOT HAND HOLDING.

SCHMIDT: WELL THESE ARE THE SMART PEOPLE>

CRAMER: INFORMATION TECHNOLOGY AND ADVERTISING.I THINK YOU HAVE REVOLUTIONIZED EVERYTHING. COULD YOU TELL ME GOLDMAN SACHS IS USING A PERCENTAGE OF THE GDP.RIGHT NOW, YOU'RE .7% OF THE GDP OF THE UNITED STATES. HERE'S WHAT I WANT TO KNOW. 600 BILLION DOLLARS IN ADVERTISING, IS IT FAIR TO THINK THAT SOMEDAY YOU WILL CAPTURE 10% OF IT.

SCHMIDT: WELL WE COULD. BY THE WAY THE NUMBER IS LARGER THAN 600 BILLION IT'S ABOUT A TRILLION GLOBALLY. IT IS PERFECTALLY POSSIBLE IN THE ONLINE WORLD IT WOULD BE HALF WE WON'T GET 100%.WE DON'T KNOW HOW LONG IT WILL TAKE, BUT WE KNOW EVERYBODY'S MOVING FROM THESE TRADITIONALLY MECHANISMS TO TARGETED AND MEASURABLE ONES AND ONLINE IS WHERE THE MEASURABLE ONES ARE.

CRAMER: RIGHT NOW PREDOMINATELY DESKTOP, BUT COUNTRIES LIKE JAPAN, MOBILE COMPUTING. IS THAT GOING TO BE UP TO SNUFF AND CAN YOU MAKE AS MUCH MONEY IN MOBILE COMPUTING BECAUSE OF THE CELL PHONE COMPANIES?

SCHMIDT: WE CAN MAKE MORE IN MOBILE THAN DESKTOP EVENTUALLY. THE REASON BECAUSE THE MOBILE COMPUTER IS MORE

TARGETED. THINK ABOUT IT YOU CARRY YOUR PHONE EVERYWHERE IT KNOWS ALL ABOUT YOU. WE CAN DO A VERY, VERY TARGETED AD.OVER TIME, WE WILL MAKE MORE MONEY FOR MOBILE ADVERTISING.

CRAMER: THERE ARE QUESTIONS THAT I'M READING, THERE'S A LOT OF THE STUFF ABOUT YOU IN THE PAPER EVERY DAY.ARE YOU COMPETING AGAINST THE ORIGINAL CONTENT PLAYERS?

SCHMIDT: WE THINK WE SEND A LOT OF TRAFFIC TO THEM. PEOPLE COME TO GOOGLE AND LOOK FOR INFORMATION AND IMMEDIATELY GO TO THE CONTENT PROVIDER. WE DONT WANT TO DISINTERMEDIATE THEM OUT, WE NEED THEIR CONTENT. WE NEED THEM TO BE SUCCESSFUL WE BUILD ADVERTISING PRODUCTS FOR THEM AND SO FORTH . VERY MUCH WE MAINTAIN THAT SEPARATION.

CRAMER: SO WHEN I READ THAT, I SHOULD-JUST THINK TWICE ABOUT WHETHER THEY ARE NOT—THERE IS A BIAS. YOU GUYS HAVE GOTTEN SO BIG, YESTERDAY, THERE WAS AN ARTICLE ABOUT HOW YOU DIDN'T HAVE GEORGIA IN THE MAP.

SCHMIDT: THAT TURNS OUT NOT TO BE TRUE.WE HAD THE SAME AMOUNT OF GEORGIA BEFORE THE WAR AS AFTER.WE'RE ADDING MORE GEORGIA GOING FORWARD CAUSE IT IS SUCH AN INTERESTING TOPIC.

CRAMER: I'VE GOT PEOPLE COMPLAINING, G-MAIL, YOU WERE OUT ON MONDAY.

SCHMIDT: THAT WAS A SCREW UP. WE FIXED THAT.WE'RE NOT PERFECT.

CRAMER: ALL RIGHT.NOW, I WANT TO TALK ABOUT PHILOSOPHICALLY, MY DAUGHTER GOT HER FIFTH GRADE ASSIGNMENT. THE FIRST THING THAT HAPPENED WAS THE TOP OF THE ASSIGNMENT, YOU ARE NOT ALLOWED TO GOOGLE IT.

SCHMIDT: REALLY?

CRAMER: YES.

SCHMIDT: ITS LIKE THE OLD THING YOU CAN'T USE THE CALCULATOR.

CRAMER: TALK ABOUT THAT.

SCHMIDT: KIDS USE IT ALL THE TIME BECAUSE IT'S A NEW WAY OF LEARNING. WHEN I WAS GROWING UP, IN VIRGINIA, THEY MADE ME MEMORIZE THE NAMES OF ALL THE CAPITALS OF EVERY COUNTY IN THE STATE. COMPLETELY USELESS INFORMATION. SO KIDS TODAY ARE GOING FROM KNOWING EVERYTHING TO BEING ABLE TO SEARCH VERY QUICKLY. THE KIDS NEED TO LEARN HOW TO SEARCH BECAUSE THEY'RE GOING TO HAVE TO SEARCH EVERYWHERE. THEY'RE GOING TO HAVE SEARCH EVERYWHERE ON DEVICES THAT THEY CARRY WITH THEM.

CRAMER: SO YOU'RE NOT WORRIED ABOUT INTELLECTUAL LAZINESS BECAUSE YOU GUYS HAVE DONE WHAT IT TOOK ME FOUR YEARS OF COLLEGE TO DO. HOW TO DO A THOROUGH SEARCH FOR PAPERS ETC

SCHMIDT: NOW, WHEN YOU WALK DOWN THE STREET, YOU SAY, WOW, IT'S INTERESTING. I'M RIDING THE TRAIN BETWEEN HERE AND D.C. AND I READ THE HISTORY OF THE TRAIN LINE.I COULD NEVER HAVE DONE THAT BEFORE.

CRAMER: A LOT OF TIMES, I THINK THAT WHAT'S HAPPENED IS THAT GOOGLE HAS BECOME SO POWERFUL AND WE'VE TAKEN IT FOR GRANTED AND WOULDN'T KNOW HOW TO DO A LOT OF THE THINGS.

SCHMIDT: I DON'T BELIEVE IN THE LAZY PEOPLE DUMB PEOPLE. I THINK PEOPLE ARE SMARTER BECAUSE THEY HAVE ACCESS TO MORE INFORMATION. GOOGLE JUST ORGANIZES IT. THE PEOPLE ARE STILL ASKING THE QUESTIONS, THEY'RE STILL THINKING IT. THEY HAVE SO MUCH MORE INFORMATION AVAILABLE TO THEM.

CRAMER: OKAY. A LOT OF THE PEOPLE FEEL THAT WITH THE 26% GROWTH YOU HIT FOR DOMESTIC THIS QUARTER THAT YOU'VE TAPPED OUT DOMESICALLY AND THAT THE GROWTH IS GOING TO HAVE TO BE INTERNATIONAL.

SCHMIDT: BY THE WAY MOST WOULD SAY 26% IS GOOD?

CRAMER: I AGREE, BUT THERE'S 31 ANALYSTS WHO SAY YOU OUGHT TO FOCUS ON THIS.

SCHMIDT: FIRST, WE DON'T KNOW WHAT'S GOING ON WITH THE GLOBAL ECONOMY. GOOGLE WILL DO BETTER IN ANY KIND OF SLOWDOWN THAN NON-TARGETED ADVERTISING BUT WE

MIGHT BE AFFECTED BY IT YOU NEVER KNOW. THE IMPORTANT POINT IS THAT OUR MODEL CONTINUES TO WORK AS PEOPLE ARE SHIFTING FROM OFFLINE TO ONLINE AND THAT SHIFT IS GOING TO HAPPEN NO MATTER WHAT.

CRAMER: DO YOU THINK WE'RE GOING TO SEE, LET ME DO THIS. GOOGLE HOME PAGE. IF WE CAN GET IT UP. I THINK YOU'VE REVOLUTIONIZED ADVERTISING. WHY CAN'T YOU SELL AS PRESENTED BY ANHEUSER-BUSCH.

SCHMIDT: WE ABSOLUTELY COULD

CRAMER: HOW MUCH WOULD PEOPLE PAY?

SCHMIDT: SOME NUMBER OF BILLIONS OF DOLLARS.

CRAMER: WHY NOT DO IT?

SCHMIDT: PEOPLE WOULDN'T LIKE IT. WE PRIORITIZE THE END USER OVER THE ADVERTISER

CRAMER: YOU'RE WILLING TO THROW AWAY A HALF A BILLION DOLLARS IN REVENUE?

SCHMIDT: ABSOLUTELY WE'RE NOT GOING TO SELL IT.

CRAMER: IF I'M A SHAREHOLDER, WHAT KIND OF ATTITUDE IS THAT?

SCHMIDT: YOU WANT THOSE USERS TO COME BACK.

CRAMER: THE DOMESTIC WOULD BE JUMPSTARTED.

SCHMIDT: EVENTUALLY, A MAGAZINE WITH ADS, PEOPLE WOULD STOP READING IT.

http://www.cnbc.com/id/26218897

* * *

611

Trial Exhibit 7326

JavaOne

Apache Harmony: An Open Invitation

Tim Ellison
Apache Harmony Project

http://harmony.apache.org

* * *

Apache's JCK issue with Sun

- First request for Java SE JCK v5 – August 2006
  - As per changes to agreement in JSPA, the "Apache Compromise"
- Sun offered JCK with FoU restrictions – 3Q2006 / 1Q2007
  - Limits how users can use compliant software, ASF reject as not allowing distribution simply under ALv2.
- Open Letter from ASF to Sun – April 2007
  - Appeal directly to Schwartz gets no response.
- ASF position on the JCP – August 2007
  - The ASF will vote "no" on any proposal involving:
  - x JCP lead who is not complying with the JSPA
  - x JSRs where the TCK-license contains FoU restrictions

Apache's JCK issue

- Current position
  - Last Sun offer with new terms for the JCK are still unacceptable.
  - Contains FoU limitations on ASF's usage of the JCK (rather than downstream user restrictions).
    - This would allow us to redistribute under ALv2.
  - However, terms contain the requirement for an 'official' notice from Sun regarding IP notice and certification requirements—effectively a *de facto* augmentation of the Apache License, and therefore unacceptable.

Apache does not make statements regarding intellectual property other than facts we explicitly know to be true and are thereby disclosed in Notice files and the Apache License.

- Oracle not yet offered alternative terms

* * *

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

Case No. 3:10-cv-03561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

v.

GOOGLE INC.,

Defendant.

Trial: May 9, 2016, 8:00 a.m.
Dept: Courtroom 8, 19th Floor
Judge: Hon. William H. Alsup

GOOGLE INC.'S DEPOSITION CLIPS OF LARRY ELLISON PLAYED BY VIDEO DURING TRIAL

Trial Exhibit 7787

---

* * *

A. Exactly, yes.

Q. —right?

Very limited internal expertise to make smart decisions. What did that refer to?

A. We'd never built a smartphone before.

Q. And didn't have the internal staff to do that?

A. Correct. No experience.

Q. Each decisions effects others and overall cost and price. That reflects the complexity; right?

A. Right.

Q. And delays have an impact on successful deliver?

A. Yeah. Every day that's delayed, Android got stronger.

* * *

Q. So obviously running on Java programming language with a Java virtual machine, that doesn't ensure success in the smartphone market; right?

* * *

THE WITNESS: Of course not. We decided not to go into the smartphone market with the Java virtual machine.

* * *

Q. BY MR. VAN NEST: Were there any other efforts that you made to enter in to this large smartphone market, apart from the various ones we've already talked about?

* * *

THE WITNESS: We never made an effort to enter the smartphone market. We analyzed whether we had a reasonable likelihood of success if we entered the smartphone market and decided, you know, that Android had already—there already was a Java smartphone with a lot of momentum, and, therefore, we could not enter the smartphone market.

Q. BY MR. VAN NEST: You've mentioned that decision or the basis for that decision a number of times. Have you seen any documents that actually reflect that a factor in your inability to get into the market was Android?

A. I haven't seen any documents that enumerate why we didn't go into the smartphone market.

Q. And you certainly haven't seen any that blame that on Android, have you?

A. As I say, I have not seen—I have not seen—again, I made the decision—it was my decision to consider the smartphone market. It was my—you know, I put it out there, so I made the decision to do the analysis, and I made the decision to kill it. And I don't know if I wrote down—I've never seen a document written down that we are not entering the smartphone market because of this reason.

* * *

Q. And did you present a model for expected Java financial performance to the board back in '09 when you made the acquisition?

A. Probably.

* * *

Trial Exhibit 9201

From: Stefano Mazzocchi <stefano@apache.org>
Sent: Monday, April 20, 2009 1:01 PM
To: members@apache.org
Subject: Re: Oracle buys Sun

Bill Stoddard wrote:

Just imagine the challenge that lies ahead of Oracle with learning to become hardware vendors in a cut throat, low margin, capital intensive business.

Unless they sell it.

The bigger question in my mind is not about hardware: what is Oracle's going to do with 500 million java-powered cell phones? What is Oracle going to do about Android's ripping off some of (now) their IP and getting away with it?

But one thing I want to say out loud: my biggest nightmares about having invested in a platform (java) that was not open was that somebody would come along and lock it down. My original fear was Microsoft, but Oracle fits the same profile.

This is why T wanted an OSI-licensed fully compliant JVM.

We forced Sun into OpenJDK as a defensive move against Harmony. If we didn't do Harmony, and OpenJDK didn't happen, all of our (and yours!) java code would feel a lot different today, wouldn't it?

But while the 'java trap' might be over as far as code goes, it is so not over about other IP, and we know very well (the OpenJDK people will realize that soon enough the hard way, I'm afraid)

Sun had but was afraid to exercise veto power at the JCP level, they knew it was a nuclear option. Would Oracle care at all if the JCP went up in the air? (with Sun and Bea gone, RedHat in pain and us religious freaks, who's left that even shows up on their radar but IBM?)

Java on the desktop is dead.

Java on the phone is dying rapidly under the attack of serious smart phones (on one side) and moore law on the other.

Java on the server is alive and well, although hardly innovative. It's enterprise legacy. It's the new COBOL, the new OS400. Oracle and IBM are strong there. Fiercefully strong.

The difference is that IBM knows (by now) how to seed and manage communities. Oracle does not (nor cares, AFAIK). The best they can do to something open they acquire is not to kill it in the process... (think Sleepycat).

Oracle MySQL, Oracle OpenOffice, Oracle NetBeans… scary.

Today is the day our (collective, painful, mostly considered foolish and laughable) investment in Harmony finally pays off.

--
Stefano.

Declarations That Are Subject to a Technical Constraint Imposed by the Java Language Specification (3rd Edition)

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.AbstractMethodError	package java.lang public class AbstractMethodError …
java.lang.ArithmeticException	package java.lang public class ArithmeticException …
java.lang.ArrayIndexOutOfBoundsException	package java.lang public class ArrayIndexOutOfBoundsException …
java.lang.ArrayStoreException	package java.lang public class ArrayStoreException …
java.lang.AssertionError	package java.lang public class AssertionError …
java.lang.ClassCastException	package java.lang public class ClassCastException …
java.lang.ClassCircularityError	package java.lang public class ClassCircularityError …
java.lang.ClassFormatError	package java.lang public class ClassFormatError …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Error	package java.lang public class Error extends Throwable
java.lang.Exception	package java.lang public class Exception extends Throwable
java.lang.ExceptionInInitializerError	package java.lang public class ExceptionInInitializerError … public ExceptionInInitializerError(Throwable thrown)
java.lang.IllegalAccessError	package java.lang public class IllegalAccessError …
java.lang.IllegalArgumentException	package java.lang public class IllegalArgumentException …
java.lang.IllegalMonitorStateException	package java.lang public class IllegalMonitorStateException …
java.lang.IncompatibleClassChangeError	package java.lang public class IncompatibleClassChangeError …
java.lang.InstantiationError	package java.lang public class InstantiationError …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.InstantiationException	package java.lang; public class InstantiationException …
java.lang.InterruptedException	package java.lang public class InterruptedException …
java.lang.LinkageError	package java.lang public class LinkageError …
java.lang.NegativeArraySizeException	package java.lang public class NegativeArraySizeException …
java.lang.NoClassDefFoundError	package java.lang public class NoClassDefFoundError …
java.lang.NoSuchFieldError	package java.lang public class NoSuchFieldError …
java.lang.NoSuchMethodError	package java.lang public class NoSuchMethodError …
java.lang.NullPointerException	package java.lang public class NullPointerException …
java.lang.OutofMemoryError	package java.lang public class OutOfMemoryError …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.RuntimeException	package java.lang public class RuntimeException extends Exception
java.lang.StackOverflowError	package java.lang public class StackOverflowError …
java.lang.UnsatisifedLinkError	package java.lang public class UnsatisfiedLinkError …
java.lang.VerifyError	package java.lang public class VerifyError …
java.lang.VirtualMachineError	package java.lang public … class VirtualMachineError …
java.lang.Deprecated	package java.lang public @interface Deprecated
java.lang.Override	package java.lang public @interface Override
java.lang.SuppressWarnings	package java.lang public @interface SuppressWarnings String[] value()
java.lang.annotation.Annotation	package java.lang public interface Annotation
java.lang.annotation.Inherited	package java.lang.annotation public @interface Inherited

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.annotation.Retention	package java.lang.annotation public @interface Retention public @interface Retention { RetentionPolicy …
java.lang.annotation.Target	package java.lang.annotation public @interface Target ElementType[] value()
java.lang.Boolean	package java.lang public … class Boolean … public boolean booleanValue()
java.lang.Byte	package java.lang public … class Byte … public byte byteValue()
java.lang.Character	package java.lang public … class Character … public char charValue() boolean isJavaIdentifierPart(int) boolean isJavaIdentifierStart(int)

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Double	package java.lang public … class Double … public double doubleValue() public … doubleToRawLong-Bits(…) public boolean isNaN() public boolean isNaN(double) public … longBitsToDou-ble(…)
java.lang.Float	package java.lang public … class Float … public float floatValue() public … float-ToRawIntBits(…) public boolean isNaN() public boolean isNaN(float) public … intBitsToFloat(…)
java.lang.Integer	package java.lang public … class Integer … public int intValue()
java.lang.Long	package java.lang public … class Long … public long longValue()
java.lang.Short	package java.lang public … class Short … public short shortValue()

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Void	package java.lang public … class Void …
java.lang.Class	package java.lang public … class Class<…> …
java.lang.ClassLoader	package java.lang public … class ClassLoader
java.lang.Cloneable	package java.lang public interface Cloneable
java.lang.Enum	package java.lang public … class Enum<…> …
java.lang.Iterable	package java.lang public interface Iterable<…> … iterator()
java.lang.Math	package java.lang public … class Math
java.lang.Object	package java.lang public class Object public final Class<?> getClass() public String toString() public boolean equals(Object obj) public int hashCode() protected Object clone() throws CloneNotSup-portedException

Class	Declaration (Partial or Full) Constrained by JLS
	public final void wait() throws IllegalMonitorStateException, InterruptedException public final void wait(long millis) throws IllegalMonitor-StateException, Inter-ruptedException public final void wait(long millis, int nanos) throws Ille-galMonitorStateException, InterruptedException public final void notify() throws IllegalMonitor-StateException public final void notifyAll() throws IllegalMonitor-StateException protected void finalize() throws Throwable
java.lang.Runtime	package java.lang public class Runtime public … exit(…) …
java.lang.String	package java.lang public ... class String ...
java.lang.System	package java.lang public … class System …
java.lang.Thread	package java.lang public … class Thread …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Thread-Group	package java.lang public class ThreadGroup …
java.lang.Throwable	package java.lang public … class Throwable …
java.io.Serializable	package java.io public interface Serializable
java.lang.annota-tion.ElementType	package java.lang.annotation public … ElementType TYPE FIELD METHOD PARAMETER, CONSTRUCTOR LOCAL_VARIABLE ANNOTATION_TYPE PACKAGE
java.lang.annota-tion.RetentionPolicy	package java.lang.annotation public … RetentionPolicy CLASS RUNTIME SOURCE

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

Case No. 3:10-cv-03561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

v.

GOOGLE INC.,

Defendant.

Trial: May 9, 2016, 8:00 a.m.
Dept: Courtroom 8, 19th Floor
Judge: Hon. William H. Alsup

ORACLE AMERICA INC.'S DEPOSITION DESIGNATIONS OF ANWAR GHULOUM PLAYED BY VIDEO DURING TRIAL

Trial Exhibit 9214

---

* * *

Q. All right. Good morning, Mr. Ghuloum. My name is Gabe Ramsey.

* * *

Q. So you understand that you've been designated today to testify on behalf of Google on a topic in a deposition notice that Oracle sent to Google?

A. Yes.

* * *

Q. Is it Google's view that the core—the APIs in the core libraries are known to Java programmers?

* * *

THE WITNESS: Yes.

* * *

Q. Do you agree that in 2007 when Android was first released, there wasn't a community of developers that knew about Android initially, correct?

A. No, there wasn't.

Q. And so Google had to attract developers to the Android platform through some means in order to get them to write programs for it?

A. Yeah. Yeah, that's true.

Q. And one of the ways that Google attracted programmers was using the Java platform, including APIs, that developers knew about?

* * *

THE WITNESS: Yeah. I mean, certainly as a part of the whole, the—the language that you're using is a—is a consideration for a developer.

BY MR. RAMSEY:

Q. Do you think that developers in 2007, application developers, by that point, there was a pretty

large community of application developers that knew about the Java APIs?

A. Yeah, but I don't know what the exact numbers are. There actually are resources for tracking this, but, yeah, it was one of the–one of the better known languages along with, you know, C, C++ and so on.

* * *

Q. Do you agree with me that the Java classes and methods that are reproduced in Android serve the same purpose in Android that they serve in Java?

* * *

THE WITNESS: Yeah, I believe they—they serve the same purpose.

* * *

BY MR. RAMSEY:

Q. Well, so, for example, if there's a class NumericShaper with—that includes methods in Android, those same methods and class are contained in Java for the same purpose?

* * *

THE WITNESS: The methods here, we strive to make sure that the methods provide the same functionality that are provided in—in OpenJDK or other implementations of Java.

BY MR. RAMSEY:

Q. All right. So the method declarations within the various Java classes contained in Android, those declarations serve the same purpose as the corresponding declarations over in the Java platform?

* * *

THE WITNESS: In terms of functionality, they provide the same functionality. That's the intent of our work.

BY MR. RAMSEY:

Q. Do you believe that the text of a method declaration in one of these Java classes in Android is likely to be understood by the—to be achieved—by developers as achieving the same purpose as the corresponding declaration in Java?

* * *

THE WITNESS: Yeah, I believe so.

* * *

Q. Do you agree that you are a senior executive in the Android organization?

* * *

THE WITNESS: I don't refer to myself as an executive, but yes, I think that would be considered the case.

* * *

Trial Exhibit 9223

Declarations That Are Subject to a Technical Constraint Imposed by the Java Language Specification (3rd Edition)

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.AbstractMethodError	package java.lang public class AbstractMethodError …
java.lang.ArithmeticException	package java.lang public class ArithmeticException …
java.lang.ArrayIndexOutOfBoundsException	package java.lang public class ArrayIndexOutOfBoundsException …
java.lang.ArrayStoreException	package java.lang public class ArrayStoreException …
java.lang.AssertionError	package java.lang public class AssertionError …
java.lang.ClassCastException	package java.lang public class ClassCastException …
java.lang.ClassCircularityError	package java.lang public class ClassCircularityError …
java.lang.ClassFormatError	package java.lang public class ClassFormatError …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Error	package java.lang public class Error extends Throwable
java.lang.Exception	package java.lang public class Exception extends Throwable
java.lang.ExceptionInInitializerError	package java.lang public class ExceptionInInitializerError … public ExceptionInInitializerError(Throwable thrown)
java.lang.IllegalAccessError	package java.lang public class IllegalAccessError …
java.lang.IllegalArgumentException	package java.lang public class IllegalArgumentException …
java.lang.IllegalMonitorStateException	package java.lang public class IllegalMonitorStateException …
java.lang.IncompatibleClassChangeError	package java.lang public class IncompatibleClassChangeError …
java.lang.InstantiationError	package java.lang public class InstantiationError …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.InstantiationException	package java.lang; public class InstantiationException …
java.lang.InterruptedException	package java.lang public class InterruptedException …
java.lang.LinkageError	package java.lang public class LinkageError …
java.lang.NegativeArraySizeException	package java.lang public class NegativeArraySizeException …
java.lang.NoClassDefFoundError	package java.lang public class NoClassDefFoundError …
java.lang.NoSuchFieldError	package java.lang public class NoSuchFieldError …
java.lang.NoSuchMethodError	package java.lang public class NoSuchMethodError …
java.lang.NullPointerException	package java.lang public class NullPointerException …
java.lang.OutofMemoryError	package java.lang public class OutOfMemoryError …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.RuntimeException	package java.lang public class RuntimeException extends Exception
java.lang.StackOverflowError	package java.lang public class StackOverflowError …
java.lang.UnsatisifedLinkError	package java.lang public class UnsatisfiedLinkError …
java.lang.VerifyError	package java.lang public class VerifyError …
java.lang.VirtualMachineError	package java.lang public … class VirtualMachineError …
java.lang.Deprecated	package java.lang public @interface Deprecated
java.lang.Override	package java.lang public @interface Override
java.lang.SuppressWarnings	package java.lang public @interface SuppressWarnings String[] value()
java.lang.annotation.Annotation	package java.lang public interface Annotation
java.lang.annotation.Inherited	package java.lang.annotation public @interface Inherited

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.annotation.Retention	package java.lang.annotation public @interface Retention public @interface Retention { RetentionPolicy …
java.lang.annotation.Target	package java.lang.annotation public @interface Target ElementType[] value()
java.lang.Boolean	package java.lang public … class Boolean … public boolean booleanValue()
java.lang.Byte	package java.lang public … class Byte … public byte byteValue()
java.lang.Character	package java.lang public … class Character … public char charValue() boolean isJavaIdentifierPart(int) boolean isJavaIdentifierStart(int)

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Double	package java.lang public … class Double … public double doubleValue() public … doubleToRawLong-Bits(…) public boolean isNaN() public boolean isNaN(double) public … longBitsToDou-ble(…)
java.lang.Float	package java.lang public … class Float … public float floatValue() public … float-ToRawIntBits(…) public boolean isNaN() public boolean isNaN(float) public … intBitsToFloat(…)
java.lang.Integer	package java.lang public … class Integer … public int intValue()
java.lang.Long	package java.lang public … class Long … public long longValue()
java.lang.Short	package java.lang public … class Short … public short shortValue()

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Void	package java.lang public … class Void …
java.lang.Class	package java.lang public … class Class<…> …
java.lang.ClassLoader	package java.lang public … class ClassLoader
java.lang.Cloneable	package java.lang public interface Cloneable
java.lang.Enum	package java.lang public … class Enum<…> …
java.lang.Iterable	package java.lang public interface Iterable<…> … iterator()
java.lang.Math	package java.lang public … class Math
java.lang.Object	package java.lang public class Object public final Class<?> getClass() public String toString() public boolean equals(Object obj) public int hashCode() protected Object clone() throws CloneNotSup-portedException

Class	Declaration (Partial or Full) Constrained by JLS
	public final void wait() throws IllegalMonitorStateException, InterruptedException public final void wait(long millis) throws IllegalMonitor-StateException, Inter-ruptedException public final void wait(long millis, int nanos) throws Ille-galMonitorStateException, InterruptedException public final void notify() throws IllegalMonitor-StateException public final void notifyAll() throws IllegalMonitor-StateException protected void finalize() throws Throwable
java.lang.Runtime	package java.lang public class Runtime public … exit(…) …
java.lang.String	package java.lang public ... class String ...
java.lang.System	package java.lang public … class System …
java.lang.Thread	package java.lang public … class Thread …

Class	Declaration (Partial or Full) Constrained by JLS
java.lang.Thread-Group	package java.lang public class ThreadGroup …
java.lang.Throwable	package java.lang public … class Throwable …
java.io.Serializable	package java.io public interface Serializable
java.lang.annota-tion.ElementType	package java.lang.annotation public … ElementType TYPE FIELD METHOD PARAMETER, CONSTRUCTOR LOCAL_VARIABLE ANNOTATION_TYPE PACKAGE
java.lang.annota-tion.RetentionPolicy	package java.lang.annotation public … RetentionPolicy CLASS RUNTIME SOURCE

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

Case No. CV 10-03561 WHA

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE INC.

Defendant.

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

ORACLE'S APRIL 3, 2012 BRIEF
REGARDING COPYRIGHT ISSUES

---

* * *

… Thus, while an application programmer does not need a license to the APIs from Oracle to author and distribute a program in the Java programming language (even if it includes calls to the APIs), whoever runs a program that includes API calls will need a license from Oracle, because that person needs an executable implementation of the APIs.

* * *

[439] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOGGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 11, 2016

Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[508] compatible or Java.

Q. During your tenure at Sun, was it permissible for a third party like that to use the APIs with their own implementation without a license from Sun?

MR. BICKS: Again, Your Honor, objection.

THE COURT: Sustained. You need to rephrase that.

MR. VAN NEST: All right.

THE COURT: That calls for a legal conclusion.

BY MR. VAN NEST

Q. Let me rephrase it.

What was the practice at Sun, during your tenure, with respect to third-party implementations of the Java APIs?

MR. BICKS: Again, Your Honor, that's beyond the scope and not fact testimony.

THE COURT: Where does that – well, where does it show up in the scope of this designation?

I think it falls within the scope. This question does fall within the scope. And so objection overruled.

Please answer the question.

THE WITNESS: Can you repeat the question.

BY MR. VAN NEST

Q. What was the practice at Sun during your tenure – during your tenure there, with respect to third parties who used the APIs with their own implementations?

A. There was nothing we could do to stop it. It was [509] completely – you know, it was fair. It's what they were – they weren't asking us to put our logo on it, and they weren't asking us to call it Java or bless or endorse it.

So we would – you know, one of the many projects that were out there that was doing exactly this, was a project supported by Oracle called Apache Harmony. And they wanted us to give them our brand. They wanted to call it Java. We kept saying no, you have to pay us a fee to do that. They didn't pay us a fee. They didn't get to use the logo.

MR. BICKS: Your Honor, this is the topic that we really talked about in the in limine that, I think, is off limits.

MR. VAN NEST: I don't believe so, Your Honor.

THE COURT: Well, I need to remind the jury of something, which is that it is – I'm not saying that what the witness says is true, not true.

I'm just giving you a fact, that the declaring code for these 37 APIs are copyrighted. And so is the structure, sequence and organization of those 37. And that comes to us as a given. Regardless of what the witness may say or not say, that is now the law in this case.

However, what the witness's attitude toward it was back at the time in question, you may consider that. But that does not change the fact that Oracle and Sun at the time did have a right, if it wished, to enforce the declaring code as a

* * *

[515] BY MR. VAN NEST

Q. Just so it's clear for all, they were using the Java APIs that were free and open; correct?

A. Yes.

MR. BICKS: Your Honor –

BY MR. VAN NEST

Q. And the implementation –

THE COURT: Wait, wait, wait. This is leading.

MR. VAN NEST: I'll withdraw the question, Your Honor. Let me just clarify.

BY MR. VAN NEST

Q. The implementing code in GNU Classpath was created by whom?

A. By people in the community who were interested in seeing a free version of Java made available.

Q. Okay. And did GNU Classpath ever take a license from Sun?

A. Absolutely not.

Q. Did Sun ever interfere with what GNU was doing with its Java implementations?

MR. BICKS: Your Honor, again, this is beyond the scope of what the Court has ruled is in play here.

THE COURT: No, this is okay. Overruled.

Go ahead. Answer the question.

THE WITNESS: No, we did not interfere with them.

[516] BY MR. VAN NEST

Q. Why not?

THE COURT: No. This is going to get into legal issues, isn't it?

MR. VAN NEST: I don't think so, Your Honor. But I understand the Court's concern. Let me move on for just a minute, and I'll ask a different question.

BY MR. VAN NEST

Q. Were you satisfied, as the CEO of Sun, that GNU's use of the Java APIs was consistent with your business practices at that time?

MR. BICKS: Again, Your Honor, objection. Leading.

THE COURT: Well, he's trying to avoid a legal problem. All right. I'll let you do it at this – overruled.

Please answer the question.

THE WITNESS: Can you repeat the question.

BY MR. VAN NEST

Q. I will try.

As the CEO, were you satisfied –

THE COURT: Instead of leading, you can say, To what extent, if at all, were you satisfied?

MR. VAN NEST: Perfect.

THE COURT: To what extent, if at all. That always solves the legal problem.

[517] BY MR. VAN NEST

Q. To what extent, if at all, were you satisfied that GNU's use of the APIs, the Java APIs, was consistent with the business practices and policies of Sun at the time?

THE COURT: But not legal. Don't get into legal issues.

BY MR. VAN NEST

Q. Business practices and policies, please, Mr. Schwartz.

A. I was annoyed, but it was completely consistent with our practices. It was competitive to what we were doing.

Q. And when you say it was consistent with your practices, what do you mean?

A. When you say open APIs will compete on implementations, it has to mean they are going to be competitive implementations. That's annoying if you're trying to get everybody to buy your product.

Q. Okay. Were there some respects which, in your view, GNU Classpath benefited Sun?

MR. BICKS: Again, Your Honor, this is leading.

THE COURT: Sustained. To what extent, if at all.

(Laughter)

BY MR. VAN NEST

Q. To what extent, if at all, Mr. Schwartz, did the activities of GNU benefit Sun?

A. They were beneficial to the extent that if you make a

* * *

[520] Q. Did – was Apache Harmony's use of Java and the APIs similar to what GNU was doing?

A. It was.

Q. Were there differences?

A. The one primary difference between products created in the Apache world versus products created in the GNU world, the GNU license, the license that you – you took when you started using their technology required you to give back any improvement you made to their code to the community.

The Apache license was very different. It was much friendlier to businesses. It said you could use whatever you want. You don't have to pay us anything. And if you embed our product, you don't have to give back anything that you're embedding with it. So it made it much more business friendly.

Q. Did Apache Harmony ever have a license from Sun?

A. No, they did not.

Q. Did Sun do anything to interfere with Apache making the Harmony product available?

A. There was nothing we could do. And there's nothing we did.

Q. And to what extent, if any, were you satisfied as CEO that Harmony's use of the Java APIs was consistent with the business practices and policies of Sun at that time?

A. Again, it was frustrating. But it was consistent with our business practices. It was a competitor to our core products, [521] one that was being promoted by other big companies. But it was not going to call itself Java, so there was nothing we could do to say you're not allowed to do that anymore.

Q. And to what extent, if any, did Apache Harmony benefit Sun?

A. Again, similar to the GNU project, it promoted the availability of Java. If you were going to pick – you know, the more products there are that are competitive, ultimately the more accessible those products will be to different types of people and different interest groups and different communities around the world.

Q. Did there come a time in your tenure at Sun, Mr. Schwartz, where Sun and Google discussed a partnership for a mobile phone?

A. Yes, we did.

Q. And did you play a role in those discussions?

A. Yes, I did.

Q. Did you personally discuss the proposal, from time to time, with Mr. Schmidt?

A. Yes, I did.

Q. And I guess you knew Mr. Schmidt – you had known Mr. Schmidt for a long time?

A. Yes.

Q. Can you tell the jury, from your perspective as CEO, what was Sun looking for in terms of a partnership with Google?

* * *

[560] evidence TX 7275_1, which is an excerpt from his announcement of open sourcing at JavaOne in 2006.

THE COURT: 7275?

MR. VAN NEST: Underscore 1, yes. It's a video.

THE COURT: Any objection?

MR. BICKS: Objection on relevance grounds, Your Honor.

THE COURT: Overruled. Go ahead.

(Government Exhibit 7275_1 received in evidence)

MR. VAN NEST: Could we play 7275_1 for the jury, please.

(Whereupon, the video was played for the jury)

BY MR. VAN NEST:

Q. Is that a somewhat younger version of Jonathan Schwartz in the video?

A. I'm not sure I'd say somewhat younger.

Q. Did Sun ever consider, during your tenure there, building a full-stack smartphone platform based on Java?

A. Absolutely, yes.

Q. Do you recall approximately when you first considered doing so?

A. I—from the earliest times surveying other handset manufacturers. We sold technology to Nokia and Ericcson and Sony and other companies.

Q. Was Sun ever able to successfully build a Java-based [561] smartphone platform?

A. We had the foundation technologies to make it work. Had Java FX Mobile, which was the core platform. But we weren't able to get it to market by the time we were sold.

Q. Why not?

A. It's complicated. It's very difficult, as Google can no doubt attest. But, you know, we also had R&D choices we had to make given R&D – Research and Development choices and staffing. Given the economic environment we were operating in, we couldn't fund every project with every dollar we had.

Q. Was Sun's failure to build its own Java smartphone platform attributable in any way to Android?

MR. BICKS: Objection, Your Honor. It's beyond the scope, the disclosure.

THE COURT: All right. Let me see the disclosure. I think I handed it back.

MR. VAN NEST: I have one in a notebook right here, Your Honor.

THE COURT: Can you highlight the language you think covers it? Highlight the language you say covers it so that I can – Mr. Van Nest, can you highlight it or circle it in some way so I can just focus on what you think is the key language.

MR. VAN NEST: Thank you, Your Honor. Thank you, Dawn.

THE COURT: All right. Have you shown counsel?

[562] MR. VAN NEST: My highlighted version, no.

THE COURT: I think the lines down near the bottom are close enough, so I'm going to allow the question.

Objection overruled.

BY MR. VAN NEST:

Q. Do you have the question in mind, Mr. Schwartz, or would you like it again?

A. Please repeat it.

Q. I will.

Was Sun's failure to build its own Java-based smartphone platform attributable in any way to the presence of Android?

A. No.

MR. VAN NEST: I pass the witness, Your Honor.

THE COURT: All right.

Cross-examination.

CROSS-EXAMINATION

BY MR. BICKS:

Q. Good morning, Mr. Schwartz.

A. Good morning.

Q. Just in terms of background, you were the CEO of Sun between 2006 to 2010?

A. That's correct.

Q. And when Oracle purchased Sun, you didn't go on to work for Oracle, did you, sir?

A. No, I did not.

* * *

[647] the exchange, and in particular the one that appears starting on page 3.

Would you please identify whether or not this is the email that you just described in your testimony as having been sent to Mr. Cizek?

A. Yes, this is what I was looking at.

Q. All right. Drawing your attention down below, where it says, "Hi, Leo." In the third paragraph of this email, would you please read the first and second sentence appearing there in that third paragraph.

A. "Right now we are moving ahead with the project, and doing an independent implementation. If Sun would like to get involved, we'd be happy to have you."

Q. And what were you seeking to communicate with Mr. Cizek at that time?

A. That I would like to partner on the development of the Java programming language for Android.

Q. All right. Let's draw your attention to the first page of Exhibit 617 and the email that goes from yourself to Mr. Cizek.

(Document displayed.)

Q. There we go.

And drawing your attention to this email, in the last paragraph, where you write, "If Sun doesn't want to partner with us to support this initiative, we are fine releasing our work and not calling it Java."

[648] Do you see that?

A. Yes, I do.

Q. What were you communicating to Mr. Cizek of Sun at the time?

A. Sure. At this point, we had already begun our internal clean-room implementation with Java. I was looking to partner with Sun to accelerate that effort and license technology. Basically, I was communicating, hey, if we can't become partners, then we're just going to go off and continue doing what we're doing.

Q. When you say "not calling it Java," does that have any relationship to the bullet we discussed in the last slide deck?

A. Yeah. I mean, obviously, if we couldn't be partners and I couldn't get a license to be able to call it Java, then I wasn't going to call it Java.

Q. Okay. All right. Let's take a look now at Exhibit 2004. Do you recognize Exhibit 2004?

A. Uhm, yeah, I do.

Q. All right. And what is it?

A. This is an email from me to Vineet Gupta, who I think was the head of business development at the time for Sun Java.

MS. ANDERSON: Your Honor, we offer Exhibit 2004 in evidence.

MS. HURST: No objection.

THE COURT: Thank you.

* * *

[694] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOGGLE, INC.,

Defendant.

San Francisco, California
Thursday, May 12, 2016

Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[737] *implementation of a Java platform and APIs?*

A. No. I mean, that wouldn't make business sense to bet our whole project on an announcement, so I had my engineers stay the course. They asked me the same questions when they read the announcement and I continued to hedge, as I had done throughout the entire project.

Q. All right. And then you said that shortly before Android was announced in November 2007, Sun actually released that OpenJDK code; is that right?

A. Yes.

MS. HURST: Objection. Leading. Also misstates the witness' testimony.

MS. ANDERSON: I'm just reorienting the witness, Your Honor.

THE COURT: I don't remember whether he said that or not. Please – I'll let you get away with it this time, but don't lead the witness on something that he has not yet testified to.

MS. ANDERSON: Yes. Yes, Your Honor. I understand.

THE COURT: All right. Overruled. Go ahead.

MS. ANDERSON: Thank you.

Q. When you learned that Sun released an open source OpenJDK Java platform shortly before the release of Android, what, if anything, did that mean to you?

A. Well, like I said, it was kind of a vote of confidence [738] that we were on the right path and they were also following the same path. So I felt pretty good about that.

But whether we could – we should switch to it or stop developing our clean room implementation, it was so close to our announcement and our launch that we didn't feel that would be helpful. It would probably actually take a little longer if we were to adopt somebody else's code rather than ship the thing we had been working on for years.

Q. Thank you.

If you could turn to Exhibit 7755. Have you seen this exhibit before?

A. Yes. It was an email between myself and Steve Horowitz.

Q. And what was the date on this email?

A. It looks like it was November 12th of 2006.

Q. Who is Steve Horowitz?

A. He was one of the engineering leaders that was running an engineering team on Android.

Q. Did Mr. Horowitz report to you?

A. Yes, he did.

MS. ANDERSON: Your Honor, we offer in evidence Exhibit 7755.

MS. HURST: Hearsay, Your Honor.

MS. ANDERSON: Your Honor, this is standard business record of what was –

THE COURT: Emails are not business records.

* * *

[803] MS. HURST: Move to admit 215.

MS. ANDERSON: No objection, Your Honor.

THE COURT: Received.

(Trial Exhibit 215 received in evidence)

BY MS. HURST:

Q. All right. Now, Mr. DeSalvo – you've broken off your talks with Sun, and Mr. DeSalvo, who was the one who had to get the Java class libraries into the platform, wrote to you: (reading)

"With talks with Sun broken off. Where does that leave us regarding Java class libraries?"

MS. ANDERSON: Objection.

BY MS. HURST:

Q. (reading)

"Ours are half-ass at best. We need another half of an ass."

MS. ANDERSON: Objection, Your Honor, to the colloquy at the beginning of the question. It lacks foundation.

THE COURT: Just a minute.

(Pause in proceedings.)

THE COURT: I don't see it is. What is the objection?

MS. ANDERSON: Your Honor, counsel made prefatory statements about – drawing collusions about the subject matter of this email before she asked a question.

THE COURT: All right. I'm going to just remind the [804] jury.

I can't interrupt every time the lawyers do it, and both sides will be guilty of this and probably have been; but when a lawyer has a little prefatory statement before a question and then asks a question, it's only what the witness says that is evidence and all that prefatory stuff, you have to just expunge it from your mind.

Because that's the easiest way that a jury can go wrong, is confusing what the lawyer says versus what the actual evidence is. You must keep that straight. The lawyers get carried away and they make speeches in front of the jury while they're asking questions. A speech is not evidence.

Both sides either have been or will be guilty of this. It applies to both sides. You must remember that.

The objection is sustained as to any prefatory statement that was made. No more prefatory statements.

BY MS. HURST:

Q. Mr. Rubin, at this point, had the talks with Sun broken off?

A. Yeah. Like four times in the past they had broken off. This is just the most recent time based on the date.

Q. You're saying you broke off talks with Sun four times before June 2006?

A. Yeah. They ebbed and flowed. We were talking to Sun in 2005.

[805] Q. All right. Mr. Rubin, at this time, you had most recently walked away from the negotiations based on control and trust issues; is that right?

A. I believe that to be accurate, yes.

Q. And Mr. DeSalvo is the person responsible for getting the Java class libraries into the Android platform; right?

A. Yes.

Q. And you no longer had that possibility of the license for that technology; true?

A. Yes.

Q. And then Mr. DeSalvo wrote to you: (reading)

"Talks with Sun broken off. Where does that leave us regarding Java class libraries? Ours are half-ass at best. We need another half of an ass."

That's what he wrote to you; right?

A. Yes.

Q. Trudy, can we see Exhibit 43.1, please.

Now, this is the exhibit that you were talking about with Ms. Anderson on direct?

A. Yes.

Q. And when we're talking about those class libraries, we're talking about these core libraries here; right?

A. Yep.

Q. And so Mr. DeSalvo had to get those core libraries into the platform, and at this point in June 2006, he didn't have [806] them; right?

A. I think the – I think the context of his email was he had about – he was about halfway done developing them.

Q. Halfway done in June 2006?

A. Yes.

Q. All right. And then you tried to get those libraries; right? You tried to get them from IBM. You tried to get them from XCE. You tried to get this them from Esmertec. You tried to get them from Skelmir. Right?

A. I was constantly looking for ways to accelerate the effort, and having people contribute to the open source effort was one of the ways I was looking for.

Q. And specifically the class libraries, those core libraries, those were what you were trying to get – part of what you were trying to get from each of those companies; right?

A. Yes.

Q. Now, you said that the discussions ended – you can take that down, Trudy – because the two

companies couldn't agree on terms; right? At this point the discussions ended because the two companies couldn't agree on terms?

A. Sorry. Is that a question?

Q. Yes. That's true; right?

A. Yes.

Q. And you personally on behalf of Google made the decision [807] that Google would walk away from that deal at that point in time?

A. Yes.

Q. And you could not agree on who would control the developer ecosystem; is that right?

A. I think that I walked away because Sun wanted to control more than I was willing for them to control.

Q. And you thought that you had to give up control as one of the key principles of an open platform; right?

A. I think with open source, the notion of control is obsolete.

Q. And you were worried that Sun had mechanisms of control that it would try to use and that would interfere with your open source strategy; right?

A. I think that's pretty accurate, yep.

Q. And you walked away from the deal because you at Google wanted to give up control of Android and just have it flourish in the open?

A. The whole idea about open source is to have very, very few restrictions on what people can do with it.

Q. You wanted to delight people with it flourishing out there in the open; isn't that right?

A. I'm not sure if I used that exact term. I may have.

THE COURT: I'll give you a hint. Anytime a lawyer turns to the jury and outstretches her arms skyward and gives a

* * *

[918] A. We talk a little bit about the name of the virtual machine, which is CLDC, and things like that.

Q. Does J2ME mean anything to you?

A. Yeah. J2ME is another term for Java ME, I believe.

Q. All right. Thank you.

And if we could just take a look at Exhibit 18 now. If you could pull that up, please.

THE COURT: You have about two more minutes.

MS. ANDERSON: We can do it, Your Honor.

Q. Exhibit 18, you were asked some questions about this exhibit and a statement in it in which you said Java.lang APIs are copyrighted around the middle of this email.

Right there in the middle Mr. Dahm under 3/24/06.

When you made this statement, what did you mean by that?

A. Well, we've been talking a lot about the declarations versus the implementation, and I was really focused on the implementation so I couldn't see how he could open source somebody else's implementation, and that's what I was commenting on.

MS. ANDERSON: I pass the witness, Your Honor. Thank you.

THE COURT: All right. Anything? Can we let the witness go.

MS. HURST: Two questions, Your Honor.

THE COURT: All right. Go ahead.

* * *

[1129] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOGGLE, INC.,

Defendant.

San Francisco, California
Monday, May 16, 2016

Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1247] …

Q. How can those persons or companies that choose to adopt the OpenJDK use those 37 Java API packages labels?

A. Well, I mentioned that OpenJDK is released with a GPL 2.0 with Classpath exception license, and because of that, companies and users of OpenJDK are able to use those 37 API package labels in all of OpenJDK, essentially in any way they wish.

And we can see on my demonstrative part of an FAQ that comes with OpenJDK, and the question here is, "Can someone create and distribute an implementation that isn't compatible with a Java specification using this code," and this code here [1248] is the OpenJDK that includes the 37 API pack labels.

And the answer to this question is, "Yes. We do not recommend or endorse that action, however. In addition, they cannot label that implementation with Java compatible or Java powered for Java ME brand and logo. These brands are your assurance that an implementation has passed the relevant TCKs."

So this shows how, because of the GPL Version 2.0 with Classpath license, that anyone can take the OpenJDK and add to it or take from it just pieces. That's what this FAQ is saying.

* * *

[1249] …

Q. Professor Astrachan, can you explain to the jury what it means to independently implement APIs?

A. Well, we've heard that phrase, and for the purposes of this explanation, we'll start with the labels from these 37 packages. And we know that, for example, Sun implemented them. And we've seen that there are other independent implementations, meaning you start from just the API method declarations and some specifications or comments about how they work and you create your own implementation without looking

at the other one. What makes them independent is that this implementation is created without looking at the other implementing code, just looking at the method declarations and the specification for how they work.

Q. How common is it to create independent implementations of APIs?

A. Well, we've seen and heard about both the GNU Classpath implementation of Java SE and the Apache Harmony implementation of Java SE. Those are two independent implementations of Java SE.

Q. Are there independent implementations of other computer languages?

A. Of other libraries, we have independent implementations. [1250] For example, the C++ libraries, there are LLVM, which is from University of Illinois and GNU software Foundation and Boost and IBM. All of those groups have created independent implementations of the C++ standard libraries.

Q. You mentioned GNU Classpath. What was GNU Classpath?

A. GNU Classpath was an independent implementation of Java SE developed by the GNU Software Foundation.

Q. And you also mentioned Apache Harmony. What is Apache Harmony?

A. Apache Harmony is an independent implementation of Java SE that was developed by the nonprofit Apache Foundation. Both the GNU Software Foundation and Apache are nonprofit foundations.

Q. To what extent, if at all, is the existence of Apache Harmony relevant to your opinions in this case?

A. Well, Apache Harmony and the other independent implementations help me understand that in general, it was very common to have independent implementations. That kind of demonstrates that it would be expected and reasonable for those things to happen.

Q. Are you aware of any examples of independent implementations of APIs by Sun?

A. Sun created Open and StarOffice, which are their own office suite of applications, kind of like Excel and Word that you might see from Microsoft. And in the OpenOffice [1251] implementation of the spreadsheet program, that uses the same functions, which are APIs, that are found in VisiCalc, one the first spreadsheet programs from the early '80s.

So functions that we see in VisiCalc for spreadsheet operations that are also part of Microsoft Excel continue to be used in StarOffice so that a spreadsheet application can run on any of those platforms using the same names and the same APIs.

Q. Are you aware of any other examples of Sun independently implementing the names of APIs?

A. Sun also has or had Solaris, an operating system that they distributed and marketed, and Solaris is a UNIX-based system. I used it when I was a graduate student. And it included some APIs from Linux, a different operating system. So those same APIs that were part of Linux were reimplemented and used as part of the Solaris operating system so that users of Solaris would be able to make the same use of those Linux APIs.

Q. Are you aware of Oracle engaging in any independent implementations of APIs?

A. Oracle distributes database products, and those database products today use SQL. I mentioned that earlier, the structured query language. SQL was something that IBM released in the mid to late '60s as part of System R. So those same APIs that are part of SQL continue to be used in database products today like the one that Oracle markets.

Q. And how, if at all, do these independent implementations [1252] support your opinions in this case?

A. All these independent implementations of APIs, from spreadsheets to databases to independent implementations of Java SE, show that it's reasonable to expect that these independent implementations was something that was common.

* * *

[1322] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOGGLE, INC.,

Defendant.

San Francisco, California
Tuesday, May 17, 2016

Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1443] …

"Because that declaring code is necessary to use the language, it is established that Google's use of the declaring code in Trial Exhibit 9223 was a fair use. This is without prejudice to evidence that other additional declaring code beyond those lines identified in Trial Exhibit 9223 either was or was not necessary for use of the Java programming language."

* * *

[1572] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOGGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 18, 2016

Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1644] …

"Q. So Nokia never actually used Java SE in a mobile phone; correct?

"A. I don't know if they every used it. My understanding is they didn't ship one.

* * *

[1694] correct?

A. I don't believe so, but I can't say that with confidence.

Q. And you recall that IBM was a big supporter of Apache Harmony; correct?

A. That sounds familiar, yes.

Q. You also testified about the SavaJE phone; correct?

A. Yes.

Q. Have you ever seen the SavaJE phone?

A. I have.

Q. It's a lot smaller than an iPhone, is it not?

A. I don't remember the relevant size, but I think that's right, yeah.

Q. The screen a lot smaller, isn't it?

A. It is.

Q. Didn't have a touchscreen?

A. I don't think so, no.

Q. In fact, the navigator on the screen had a little joy stick; right?

A. Yes, like most smartphones in that era. That's true.

Q. It didn't have a physical typewriter keyboard like the T-Mobile Sidekick did; right?

A. I don't recall.

Q. Just has a phone keyboard?

A. I don't recall exactly, no.

Q. Do you recall that to text message you would have to press [1695] the letter number two, three times to get a C up there?

A. Yeah. That's how all those phones in that era work, with 9 – 9-key keypads.

Q. You would say it's very different than the modern smartphone, wouldn't you, sir?

A. It is.

* * *

[1767] packages were in smartphones and they were in feature phones. And the competition occurred in both. However, various people have defined feature phones and smartphones. So I just don't think the distinction is that important.

Q. And can you give us any examples of where Java was used in smartphones?

A. Yes. You've heard testimony about this. Mr. Rubin talked about Danger and Sidekick, which was a smartphone that used Java.

We've also heard talk about the SavaJe phone, which was another smartphone that used Java.

Q. Any other examples that you can think of?

A. I think BlackBerry, in the early days, was a Java licensee. And its phones at the time were considered smartphones.

Q. And when you hear the phrase "smartphone" to you, in your work here, what is a smartphone?

A. Well, it depends a little bit on the time period. But basically it's – it's a phone which typically has a color touchscreen, has a full keyboard, and has, you know, other capabilities that consumers desire.

Q. And what was the name of the first Android device? And when did it come out?

A. So the first Android device was the HTC Dream, which came out in the fall of 2008.

* * *

[1769] economic perspective, they both contained at least portions of the copyrighted declarations and SSO that are at issue here.

So in terms of substitution, what I'm looking for is substitution for any of the opportunities that Sun and then Oracle had to license their intellectual property however they may have chosen to arrange that intellectual property into different versions.

Q. And do you have an example of a product that transitions from ME to SE?

A. Yes. My understanding is that with respect to the Amazon Kindle, this was initially licensed by Amazon based on Java ME. As the Kindle has evolved and become more powerful, Amazon now has a license that also covers SE.

Q. Now, let's just, then, sum up this transformative question.

In your opinion, from an economic perspective, how transformative, if at all, was Google's use of the 37 Java API packages in Android?

A. So, again –

MR. VAN NEST: Object, Your Honor, to the use of the term "transformative" in this setting.

THE COURT: Well, the jury will keep in mind the witness says that transformative was not an economic term. But he did say that he could give an opinion on the issue of substitution.

* * *

[1786] …

You had access to all of the deposition videos and transcripts that our jurors have seen played during the trial; right?

A. Yes.

Q. In fact, your report says you had dozens of depositions that you were free to read?

A. Yes.

Q. And hundreds of documents you were free to review?

A. Yes.

Q. Right?

And among those were the sworn testimony of Mr. Gering, at Sun, that Sun was never able to turn the SavaJe into a commercial product; right?

A. Yes.

Q. And you – you gave SavaJe as a great example of a Java SE-licensed product, in your direct examination, didn't you?

A. I don't think I said it was a great example. I just said it was an example.

Q. It was an example of failure, though; right?

A. It was not a successful product.

[1787] Q. That's right.

In fact, it never made it to the market in any significant way; right, Dr. Jaffe?

A. That is correct.

Q. And you were calling it a smartphone.

By the way, I notice none of the Oracle experts have actually shown us the SavaJe, including you; right?

A. That is correct.

Q. Do you recognize this (indicating)?

A. Could I see it?

Q. Sure.

MR. VAN NEST: May I approach the witness, Your Honor?

THE COURT: Yes.

BY MR. VAN NEST

Q. That's a SavaJe phone –

A. Yes, it is.

Q. – right?

A. Yes, it is.

Q. No touchscreen?

A. That's correct.

Q. No keypad – no QWERTY?

A. That is correct.

THE COURT: Better explain to the jury what you mean by QWERTY.

MR. VAN NEST: QWERTY, the keyboard like on a – oh, [1788] I'm sorry. I'm not the witness.

(Laughter)

BY MR. VAN NEST

Q. Go ahead, Dr. Jaffe.

A. I like being a witness.

So if you picture a computer or a typewriter keyboard, starting with your pinkie it goes Q, W, E, R, T, Y. So people sometimes make that into a word. And they call it QWERTY.

There's actually an interesting historical story about it.

THE COURT: All right. That's too much.

(Laughter)

THE COURT: Next question.

MR. VAN NEST: We all get in trouble.

BY MR. VAN NEST

Q. In any event, Dr. Jaffe, this is the SavaJe phone that you and others have said was a great early version of smartphone; right?

A. I never said it was great.

Q. Okay. Fair enough.

Now, folks internal at Sun did discuss the smartphone and the feature phone markets as separate things, didn't they?

A. They did on occasion, yes.

Q. And on occasion they drew a big distinction between the smartphone market and the feature phone market; right?

A. They drew distinctions, yes.

* * *

[1810] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Thursday, May 19, 2016

Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1861] know, revenue is going up and Java SE is doing fine; right?

A. I don't know whether it's going up or down. I know that they do continue to license it in those markets.

Q. Okay. Let's actually play some testimony from Donald Smith, who was a 30(b)(6) witness, as his Honor explained, from his November 20, 2015 deposition at page 277, lines 23 to 278, line 7.

(Whereupon, the video was played for the jury)

BY MR. VAN NEST:

Q. Did you talk to anybody in the Java SE business at Oracle America to confirm what Mr. Smith said?

A. No.

Q. So even though you knew that the copyrighted work was Java SE, you didn't go to the trouble of talking to the Java SE managers to see how that was doing?

A. No.

Q. Did you have Keystone do that?

A. No.

Q. Did anybody do that?

A. Not that I know of.

Q. So as far as you know, what Mr. Smith says is right? They're doing just fine?

A. As far as I know, yes.

Q. Okay. And Mr. Ellison also testified that the Java business overall was doing fine; right?

[1862] A. Yes.

Q. He said that in the video we played, that as far as he knew, they were hitting their targets and Java SE – the Java business overall was doing fine. That's what he said; right?

A. Yes.

Q. By the way, yesterday, Dr. Jaffe, you mentioned the HTC Touch Pro as a Java smartphone, I think you said.

A. Yes.

Q. Now, actually HTC Touch Pro runs Windows mobile platform from Microsoft; right?

A. But it has Java on it as well, yes.

Q. Let's break this down. You know that Android is a full stack, top to bottom, it does everything a smartphone needs to do; right?

A. Yes.

Q. There's no Java product that does that?

A. That is correct.

Q. Java provides some functionality, but it needs a much bigger system around it to be relevant; right?

A. Provides a portion – it provides the applications interface in a larger system, yes.

Q. So when you talked about smartphones using Java, you didn't have any examples, other than maybe SavaJe, where the whole thing was Java; right?

A. Of course.

* * *

[1914] supplied to you?

A. Yes, it was.

Q. So you knew that that testimony was in there before you came and testified today that these were not competitive products; isn't that true?

A. Yes. As it turns out, Sun did not have a competitive product.

Q. Well, Mr. Rubin was certainly worried about one, wasn't he?

A. Uhm, I don't know. He was worried about them being upset that they weren't going to partner with – with Sun.

Q. Let me ask you this, Dr. Leonard: Did you see exhibit 5322, that was just displayed during Dr. Jaffe's testimony, where Rich Miner – he was one of the founders of Android; right?

A. Yes, I think so.

Q. – said, "As a case in point, if we were not doing what we are doing SavaJE would have probably gotten more funding."

Now, you know SavaJE was a full stack operating system with SE in it, don't you, sir? You know that?

A. Well, I wouldn't go so far as to say that it was a smartphone operating system that would have done what a smartphone could do.

But it was a full stack operating – or it had a phone with a full stack operating system on it.

* * *

678

Trial Exhibit 1026

SUN PROPRIETARY/CONFIDENTIAL:
INTERNAL USE ONLY

USA Sun Microsystems, Inc.
Document Separator Sheet

- Agreement Type: (40A/N)
  - __ Confidential Disclosure
  - XX Sales Outbound
  - __ Inbound Technology
  - __ Marketing
  - __ Supply/Vendor
- Name of Third Party: (50A/N) Danger Inc.
- Name of Sun Entity: (50A/N)
  - XX Sun Microsystems, Inc.
  - __ Sun Federal
  - __ BV1
  - __ Techco
- Agreement Effective Date:
  (9A/N) (DD MMM YYYY) 26 AUG 2003
- Agreement Expire/Term Date:
  (9A/N) (DD MMM YYYY) N/A
- Agreement Registry ID/Other #: (40A/N) 101560
- Additional Search Criteria: (50A/N) CLDC SCSL

Location: (ISO Country Code) (2+20A/N)
US + SAN JOSE

Iron Mountain Box #: (9A/N) 133510277

Agreement Negotiator/Legal Contact: (20A/N)
NADAN, UTRIS

* * *

GLOSSARY

1. "Commercial Use" means any use (excluding Internal Deployment Use) or distribution, directly or indirectly of Compliant Covered Code by You to any third party, alone or bundled with any other software or hardware, for direct or indirect commercial or strategic gain or advantage, subject to execution of Attachment D by You and Original Contributor.

2. "Community Code" means the Original Code, Upgraded Code, Error Corrections, Shared Modifications, or any combination thereof.

3. "Community Webserver(s)" means the webservers designated by Original Contributor for posting Error Corrections and Shared Modifications.

4. "Compliant Covered Code" means Covered Code that complies with the requirements of the TCK.

5. "Contributor" means each Licensee that creates or contributes to the creation of any Error Correction or Shared Modification.

6. "Covered Code" means the Original Code, Upgraded Code, Modifications, or any combination thereof.

7. "Error Correction" means any change made to Community Code which conforms to the Specification and corrects the adverse effect of a failure of Community Code to perform any function set forth in or required by the Specifications.

8. "Executable" means Covered Code that has been converted to a form other than Source Code.

9. "Extension(s)" means any additional classes or other programming code and/or interfaces developed

by or for You which: (i) are designed for use with the Technology; (ii) constitute an API for a library of computing functions or services; and (iii) are disclosed to third party software developers for the purpose of developing software which invokes such additional classes or other programming code and/or interfaces. The foregoing shall not apply to software development by Your subcontractors to be exclusively used by You.

10. "Intellectual Property Rights" means worldwide statutory and common law rights associated solely with (i) patents and patent applications; (ii) works of authorship including copyrights, copyright applications, copyright registrations and "moral rights"; (iii) the protection of trade and industrial secrets and confidential information; and (iv) divisions, continuations, renewals, and re-issuances of the foregoing now existing or acquired in the future.

11. "Internal Deployment Use" means use for productive purposes of Compliant Covered Code (excluding Research Use) within Your business or organization only by Your employees and/or agents, subject to execution of Attachment C by You and Original Contributor, if required.

12. "Licensee" means any party that has entered into and has in effect an agreement with Original Contributor granting substantially similar rights as the Sun Community Source Code License Version 2.25.

13. "Modification(s)" means (i) any change to Covered Code; (ii) any new file or other representation of computer program statements that contains any portion of Covered Code; and/or (iii) any new Source Code implementing any non-trivial portion of the Specifications, provided that this subsection (iii) does

not include an implementation of a portion of the Specifications not specific to the Technology or Java and in common usage (or a combination of such common usage portions of the Specifications provided such combination is not specific to the Technology or Java and in common usage), that is independently developed without the use of the Specifications or Covered Code.

14. "Original Code" means the initial Source Code for the Technology as described on the Technology Download Site.

15. "Original Contributor" means Sun Microsystems, Inc., its affiliates and its successors and assigns.

16. "Reformatted Specifications" means any revision to the Specifications which translates or reformats the Specifications (as for example in connection with Your documentation) but which does not alter, subset or superset the functional or operational aspects of the Specifications.

17. "Research Use" means use and distribution of Covered Code only for Your research, development, testing, educational or personal and individual use, and expressly excludes Internal Deployment Use and Commercial Use.

18. "SCSL Webpage" means the Sun Community Source license webpage located at http://sun.com/software/communitysource, or such other url that Sun may designate from time to time.

19. "Shared Modifications" means Modifications provided by You, at Your option, pursuant to Section 2.2, or received by You from a Contributor pursuant to Section 2.3.

20. "Source Code" means computer program statements written in any high-level, readable form suitable for modification and development.

21. "Specifications" means the specifications for the Technology and other documentation, as designated on the Technology Download Site, as may be revised by Original Contributor from time to time.

22. "Sun Trademarks" means Original Contributor's SUN, JAVA, and JINI trademarks and logos, whether now used or adopted in the future.

23. "Technology" means the technology described in Attachment B, and Upgrades.

24. "Technology Compatibility Kit" or "TCK" means the test programs, procedures and/or other requirements, designated by Original Contributor for use in verifying compliance of Covered Code with the Specifications, in conjunction with the Original Code and Upgraded Code. Original Contributor may, in its sole discretion and from time to time, revise a TCK to correct errors and/or omissions and in connection with Upgrades.

25. "Technology Download Site" means the site(s) designated by Original Contributor for access to the Original Code, Upgraded Code, TCK and Specifications.

26. "Upgrade(s)" means new versions of Technology designated exclusively by Original Contributor as an "Upgrade" and released by Original Contributor from time to time.

27. "Upgraded Code" means the Source Code for Upgrades, possibly including Modifications made by Contributors.

28. "You(r)" means Danger, Inc., 124 University Avenue, Palo Alto, CA 94301.

ATTACHMENT A
REQUIRED NOTICES

ATTACHMENT A-1
REQUIRED IN ALL CASES

'The contents of this file, or the files included with this file, are subject to the current version of Sun Community Source License for [fill in name of applicable Technology] (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at http://sun.com/software/communitysource. See the License for the rights, obligations and limitations governing use of the contents of the file.

The Original and Upgraded Code is [fill in name and version of applicable Technology]. The developer of the Original and Upgraded Code is Sun Microsystems, Inc. Sun Microsystems, Inc. owns the copyrights in the portions it created. All Rights Reserved.

Contributor(s):_____[SAMPLE]_____

Associated Test Suite(s) Location:_____[SAMPLE]____

ATTACHMENT A-2
SAMPLE LICENSEE CERTIFICATION

'By clicking the 'Agree' button below, You certify that You are a Licensee in good standing under the Sun Community Source License, [fill in applicable Technology and Version] ("License") and that Your access, use and distribution of code and information You may obtain at this site is subject to the License."

ATTACHMENT A-3
REQUIRED STUDENT NOTIFICATION

"This software and related documentation has been obtained by your educational institution subject to the Sun Community Source License, [fill in applicable Technology]. You have been provided access to the software and related documentation for use only in connection with your course work and research activities as a matriculated student of your educational institution. Any other use is expressly prohibited.

THIS SOFTWARE AND RELATED DOCUMENTATION CONTAINS PROPRIETARY MATERIAL OF SUN MICROSYSTEMS, INC, WHICH ARE PROTECTED BY VARIOUS INTELLECTUAL PROPERTY RIGHTS.

You may not use this file except in compliance with the License. You may obtain a copy of the License on the web at http://sun.com/software/communitysource."

ATTACHMENT B

Java(tm) 2 Platform, Micro Edition, Connected Limited Device Configuration 1.1 and Mobile Information Device Profile 2.0 Source Technology

Description of "Technology"

Java(tm) 2 Platform, Micro Edition, Connected Limited Device Configuration 1.1 and Mobile Information Device Profile 2.0 Source Technology as described on the Technology Download Site.

ATTACHMENT C
INTERNAL DEPLOYMENT USE

This Attachment C is only effective for the Technology specified in Attachment B, upon execution of Attachment D (Commercial Use License) including the requirement to pay royalties. In the event of a conflict between the terms of this Attachment C and Attachment D, the terms of Attachment D shall govern.

1. Internal Deployment License Grant. Subject to Your compliance with Section 2 below, and Section 8.10 of the Research Use license; in addition to the Research Use license and the TCK license, Original Contributor grants to You a worldwide, non-exclusive license, to the extent of Original Contributor's Intellectual Property Rights covering the Original Code, Upgraded Code and Specifications, to do the following:

a) reproduce and distribute internally, Original Code and Upgraded Code as part of Compliant Covered Code, and Specifications, for Internal Deployment Use,

b) compile such Original Code and Upgraded Code, as part of Compliant Covered Code, and reproduce and distribute internally the same in Executable form for Internal Deployment Use, and

c) reproduce and distribute internally, Reformatted Specifications for use in connection with Internal Deployment Use.

2. Additional Requirements and Responsibilities. In addition to the requirements and responsibilities described under Section 3.1 of the Research Use license, and as a condition to exercising the rights

granted under Section 3 above, You agree to the following additional requirements and responsibilities:

2.1 Compatibility. All Covered Code must be Compliant Covered Code prior to any Internal Deployment Use or Commercial Use, whether originating with You or acquired from a third party. Successful compatibility testing must be completed in accordance with the TCK License. If You make any further Modifications to any Covered Code previously determined to be Compliant Covered Code, you must ensure that it continues to be Compliant Covered Code.

ATTACHMENT D: COMMERCIAL USE LICENSE (CLDC CHNLS) Danger, Inc.

1. Effect. This Attachment D is to this Sun Community Source License for Java 2 Platform, Micro Edition Connected Limited Device Configuration and Mobile Information Device Profile ("SCSL"). You have agreed to the terms of the SCSL by executing a hardcopy SCSL (attached hereto) with Original Contributor. You acknowledge that the SCSL is binding on You. This Attachment D is effective only if signed below by You and Original Contributor, and applies to Your Commercial Use of Original Code and Upgraded Code. All capitalized terms used herein shall have the same meaning set forth in the SCSL, unless otherwise stated.

2. Term. The SCSL (including this Commercial Use license) shall have an initial term of 3 years and shall automatically renew for additional one year terms unless either party provides notice to the other no less than 60 days prior to an anniversary date.

3. Commercial Use License Grant. Subject to Your compliance with Sections 4 and 7 below, Section 8.10 of the Research Use license, and the TCK license and Section 2 of Attachment F; in addition to the Research Use license, the TCK license, and the Internal Deployment Use license, Original Contributor grants to You a worldwide, non-exclusive, non-transferable license, to the extent of Original Contributor's Intellectual Property Rights covering the Original Code, Upgraded Code and Specifications, to do the following within the specified field of use in Section 7:

a) reproduce and distribute Compliant Covered Code;

b) compile Compliant Covered Code and reproduce and distribute the same in Executable form through multiple tiers of distribution; and

c) reproduce and distribute Reformatted Specifications in association with Compliant Covered Code.

4. Additional Requirements and Responsibilities. In addition to the requirements and responsibilities specified in the Research Use license, the TCK license and the Internal Deployment license, and as a condition to exercising the rights granted in Section 3 above, You agree to the following additional requirements and responsibilities:

a) Distribution of Source Code. Source Code of Compliant Covered Code may be distributed only to a Customer (as defined in Section 7(a) below) who is at all times a Code Sharing Licensee, defined as: (i) a Licensee of the same Technology; or (ii) a party otherwise authorized by Original Contributor in a written license agreement to receive and share Covered Code, provided that the

Customer Agreement (as defined in Section 7(1) below) with such party makes clear that Compliant Covered Code provided to such party is subject to the restrictions and obligations of such party's license agreement with Original Contributor.

(i) You must not enter into any agreements purporting to put Covered Code in escrow, unless the party to whom such code would be released is at the time of release a Licensee of the same Technology ("Releasee"). In such cases, notwithstanding Section 4a) above, You may place the code with an escrow agent that is not a Licensee, provided that You contractually require in a written agreement and ensure that (i) except as permitted in "Verification of Escrow Materials" below, the escrow agent cannot and does not copy, modify, view, transfer (except to the Releasee in compliance herewith) or in any way use the Covered Code at any time, (ii) the escrow agent and the escrowed Covered Code are at all times located in the United States, and (iii) the escrow agent releases copies of the Covered Code only with 10 days prior written notice to Original Contributor and only to a party that the Original Contributor has confirmed in writing to the escrow agent is a Licensee of the same Technology whose License is valid and current at the time of proposed release. Original Contributor shall not unreasonably withhold or delay such confirmation. In addition, You must provide prior written notice to Original Contributor of the name and address of any escrow agent to which Covered Code is provided, along with a copy of the agreement between You and the escrow agent and the name of each potential Releasee. In Your

agreement with the escrow agent, Original Contributor shall be named as an intended third party beneficiary of the contractual requirements required above (or the escrow agent must provide to Original Contributor such escrow agent's binding written agreement to those contractual requirements), and in the event of a breach of such requirements by the escrow agent, You shall indemnify Original Contributor for all out of pocket costs and expenses, including attorneys fees, incurred in enforcing the agreement and Original Contributor's rights. All escrow related costs and fees shall be borne by You.

(ii) Verification of Escrow Materials. If the escrow agent is a Licensee, it may copy and view the Covered Code to perform limited verification of the Covered Code, in accordance with the escrow agreement, to determine that the Covered Code compiles and runs properly. The escrow agent may also hire a third party independent contractor to perform verification of the Covered Code, in accordance with the escrow agreement, to determine that it compiles and runs properly, if such third party is a Licensee.

b) Distribution of Executable Code. You may distribute the Executable version(s) of Compliant Covered Code under a license of Your choice, provided that (i) You are in compliance with the terms of this License, including without limitation Section 7(l), below, and (ii) You make it absolutely clear that Your license is offered by You alone, not by Original Contributor or any other Contributor.

c) Branding. Products integrating Compliant Covered Code used for Commercial Use may be

branded with the Technology compliance logo under a separate trademark license required to be executed by You and Original Contributor concurrent with execution of this Attachment D.

d) Integration with Configuration. The Java 2 Platform, Micro Edition, Mobile Information Device Profile ("MIDP") and Java 2 Platform, Micro Edition, Connected Limited Device Configuration ("CLDC") implementations must be tightly integrated with, and must be configured to run in conjunction with, each other, and each must meet Original Contributor's compatibility requirements in accordance with this License.

5. Indemnity/Limitation of Liability. The provisions of Section 7.1 of the Research Use license are superseded by the following:

a) Your Indemnity Obligation. You hereby agree to defend, at Your expense, any legal proceeding brought by a third party against Original Contributor or any Licensee to the extent it is based on a claim: (i) that the use, reproduction or distribution of any of Your Error Corrections or Shared Modifications is an infringement of a third party trade secret or a copyright in a country that is a signatory to the Berne Convention; (ii) arising in connection

* * *

8. Notice of Breach or Infringement. Each party shall notify the other immediately in writing when it becomes aware of any breach or violation of the terms of this License, or when You become aware of any potential or actual infringement by a third party of the Technology or Original Contributor's Intellectual Property Rights therein.

9. Proprietary Rights Notices. You may not remove any copyright notices, trademark notices or other proprietary legends of Original Contributor or its suppliers contained on or in the Original Code, Upgraded Code and Specifications.

10. Notices. All written notices required by this License must be delivered in person or by means evidenced by a delivery receipt and will be effective upon receipt by the persons at the addresses specified below.

Original Contributor:	You:
Sun Microsystems, Inc.	Danger, Inc.
4150 Network Circle,	124 University Avenue
Santa Clara, California, USA 95054	Palo Alto, CA 94301
Attn.: VP, Software Sales	Attn.: General Counsel
cc: Legal (Software Sales)	

11. Disclaimer of Agency. The relationship created hereby is that of licensor and licensee and the parties hereby acknowledge and agree that nothing herein shall be deemed to constitute You as a franchisee of Original Contributor. You hereby waive the benefit of any state or federal statutes dealing with the establishment and regulation of franchises.

12. Confidentiality. The parties shall keep and maintain in confidence the terms and conditions of this Attachment D except to the extent disclosure is required by law.

13. No Third Party Beneficiaries. Nothing herein creates third party beneficiary rights of any kind in favor of any party other than Original Contributor.

AGREED TO AND ACCEPTED BY:

You:	Original Contributor:
Danger, Inc.	Sun Microsystems, Inc.
By: /s/ Henry Nothhart	By: /s/ Neal Civjan
Name: Henry R. Nothhart	Name: Neal Civjan
Title: Chairman and CEO	Title: Vice President WW OEM Software Sales Global Sales Operations
Date:8/13/03	Date:8/26/03

ATTACHMENT E
TECHNOLOGY COMPATIBILITY KIT

The following license is effective for the Java(tm) 2 Platform, Micro Edition Connected Limited Device Configuration Technology Compatibility Kit and the Java(tm) 2 Platform, Micro Edition Mobile Information Device Profile Technology Compatibility Kit, but only upon execution of a separate support agreement between You and Original Contributor (subject to an annual fee) as described on the SCSL Webpage. The Technology Compatibility Kit for the Technology specified in Attachment B may be accessed at the Technology Download Site only upon execution of the support agreement.

1. TCK License.

a) Subject to the restrictions set forth in Section 1.b below and Section 8.10 of the Research Use license, in

* * *

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

Case No. 3:10-cv-03561 WHA

ORACLE AMERICA, INC.,

Plaintiffs,

v.

GOOGLE INC.,

Defendant.

Trial Date: May 9, 2016

Dept: Courtroom 8, 19th Fl.

Judge: Hon. William Alsup

GOOGLE INC.'S DEPOSITION CLIPS OF LARRY ELLISON PLAYED BY VIDEO DURING TRIAL

Trial Exhibit 7787

* * *

08/12/2011

* * *

A. Exactly, yes.

Q. – right?

Very limited internal expertise to make smart decisions. What did that refer to?

A. We'd never built a smartphone before.

Q. And didn't have the internal staff to do that?

A. Correct. No experience.

Q. Each decisions effects others and overall cost and price. That reflects the complexity; right?

A. Right.

Q. And delays have an impact on successful deliver?

A. Yeah. Every day that's delayed, Android got stronger.

Q. So obviously running on Java programming language with a Java virtual machine, that doesn't ensure success in the smartphone market; right?

THE WITNESS: Of course not. We decided not to go into the smartphone market with the Java virtual machine.

Q. BY MR. VAN NEST: Were there any other efforts that you made to enter in to this large smartphone market, apart from the various ones we've already talked about?

THE WITNESS: We never made an effort to enter the smartphone market. We analyzed whether we had a reasonable likelihood of success if we entered the 00136:01 smartphone market and decided, you know, that Android had already – there already was a Java smartphone with a lot of momentum, and, therefore, we could not enter the smartphone market.

Q. BY MR. VAN NEST: You've mentioned that decision or the basis for that decision a number of times. Have you seen any documents that actually reflect that a factor in your inability to get into the market was Android?

A. I haven't seen any documents that enumerate why we didn't go into the smartphone market.

Q. And you certainly haven't seen any that blame that on Android, have you?

A. As I say, I have not seen – I have not seen – again, I made the decision – it was my decision to consider the smartphone market. It was my – you know, I put it out there, so I made the decision to do the analysis, and I made the decision to kill it. And I don't know if I wrote down – I've never seen a document written down that we are not entering the smartphone market because of this reason.

Q. And did you present a model for expected Java financial performance to the board back in '09 when you made the acquisition?

A. Probably.

Q. How has the Java portion of your business performed financially against that model in '09 and '10 and so far in 2011?

THE WITNESS: I think – again, I think it was a reasonably conservative board model. And I think it's done okay, but I'd have to look at the numbers. I don't know offhand.

Q. BY MR. VAN NEST: Has Java revenue increased since you acquired Sun?

A. I believe so.

Q. By what kind of factor? Do you know?

A. I don't know.

Q. Do you know why it's increased?

A. We – we're selling more. What do you mean by that? We're – we're selling – you know, we're 00152:01 selling more of it.

Q. So you're doing better in the categories you have?

A. Right.

Q. And you're expanding to the categories as well?

A. Yeah.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

Case No. 3:10-cv-03561 WHA

ORACLE AMERICA, INC.,

Plaintiffs,

v.

GOOGLE INC.,

Defendant.

Trial Date: May 9, 2016

Dept: Courtroom 8, 19th Fl.

Judge: Hon. William Alsup

GOOGLE INC.'S DEPOSITION CLIPS OF DONALD SMITH PLAYED BY VIDEO DURING TRIAL

Trail Exhibit 7788

* * *

11/20/2015

* * *

Q. Okay. Good morning, Mr. Smith.

A. Good morning.

Q. Could you please state your full name for the record?

A. Sure. It's Donald Owen Smith.

Q. Have you ever been deposed before?

A. I have not.

Q. Well, I assume you understand that it's a general question and answer process?

A. I do.

Q. I'll be asking the questions; you'll be answering them. If you don't understand any of my questions, please just feel free to let me know. I'm happy to restate it in a way that you can understand.

A. Okay.

Q. If you don't say that, if you don't ask me to clarify, I'll just assume you understood the question.

A. Sure.

Q. And you understand that you're here as witness; is that correct?

A. I do.

Q. And you understand that that means you're speaking on behalf of Oracle, the company?

A. I do.

Q. So do you understand the Java language to include the APIs?

THE WITNESS: Yes. I mean, the APIs are a critical part of the Java language.

Q. Would you say that's true for the APIs that are at issue in the case?

THE WITNESS: Yes, those APIs are a fundamental part of what makes Java Java – what makes a developer recognize Java.

Q. Do you believe that the Java language and the Java APIs are inseparable?

THE WITNESS: Inseparable? I'm sorry, I'm not sure I fully understand the context of that.

BY MR. MULLEN:

Well, in other words, can the two be separated for purposes of what use is permissible?

THE WITNESS: So again, like, as it relates to the specification and both the languages as it's defined, no, they're not separable. It's all defined together under the same specification.

Q. Is one of the goals of your job to encourage developers to use the Java programming language?

THE WITNESS: Yes.

Is it part of Oracle's overall business strategy, as it relates to Java, to encourage people to write in the Java programming language?

THE WITNESS: Yes, we – again, a lot of our products are based on Java and are based on companies building Java applications. So the more that the platform is available, the better it is for us as a business.

Q. Do you have an understanding of how many Java developers there are in the world today, roughly?

THE WITNESS: Yes. So the approved number that we use is 10 million plus. There's often press and analysts that talk about it in terms of tens of millions.

BY MR. MULLEN:

Q. Do you have an understanding of whether that number of developers has increased over time?

THE WITNESS: It has been increasing over time.

BY MR. MULLEN:

Q. Has it increased since 2011?

THE WITNESS: I would – yes, it would be fair to say it's increased since 2011.

Q. Do you expect Java SE revenue to increase going forward?

THE WITNESS: So when you say "Java SE revenue," it depends on what you mean, right, because there are a number of ways we generate revenue from Java SE. Some are growing; some are not.

BY MR. MULLEN:

Q. How is the business doing overall?

THE WITNESS: Right. So Java SE – like Java SE Advanced is growing well. Support revenue is growing well. So that the Java SE business – I mean, I would have to have the actual accounting numbers here, but it's like $150 million for what I personally carve out as what my team is responsible for generating.

Q. Mr. Smith, do you recall that you testified earlier today about the relationship between the Java language and the Java API?

A. Yes.

Q. And do you recall that you said that the APIs are a critical part of the Java language?

A. Yes.

Q. Do you believe that statement was accurate?

A. In the context of the specification, that's what I was referring to, but no, I don't believe it was accurate in isolation.

Q. Did you speak with Mark Reinhold to get a correct understanding of the relationship between the Java language and the Java API?

A. Yes. I've spoken with Mark Reinhold. He's the expert in how the specification is created and how everything is created and interconnects.

Q. And are there others at Oracle who are more knowledgeable about the technical details of the Java language or the Java API than you?

THE WITNESS: Well, so Mark Reinhold would be particular expert in this topic.

Q. And is the Java language within your job responsibilities?

THE WITNESS: It would be under Mark.

At the time you answered the questions this morning, what did you understand the context of those questions to be?

THE WITNESS: So the context that I was working on was talking about within the specification itself, like the specification in its entirety.

BY MS. LEWIS-GRUSS:

Q. So was it – so now do you understand the context – do you understand that you were asked a different question this morning?

THE WITNESS: Yeah, so this morning I was concerned that I misunderstood the question, and that's why I asked if it would be okay to talk to Mark Reinhold to further my knowledge in this area, in this topic.

Q. So if I asked you whether or not the Java API and the Java language were inseparable, what would your answer be?

THE WITNESS: Yeah, so they are separate documents under the same specification, and they are separate.

Q. As you sit here today, do you have personal knowledge of the distinction between the Java language and the Java API?

THE WITNESS: I'm sorry, I'm just not sure I understand the question.

Q. And when I asked you those questions this morning, were you confused at that time?

A. I was confused, yes.

Trial Exhibit 1045

The Apache Software Foundation
Blogging in Action

The Apache Software Foundation Blog

Thursday Dec 09, 2010

The ASF Resigns From the JCP Executive Committee

The Apache Software Foundation has resigned its seat on the Java SE/EE Executive Committee. Apache has served on the EC for the past 10 years, winning the JCP "Member of the Year" award 4 times, and recently was ratified for another term with support from 95% of the voting community. Further, the project communities of the ASF, home to Apache Tomcat, Ant, Xerces, Geronimo, Velocity and nearly a 100 mainstay java components have implemented countless JSRs and serve on and contribute to many of the JCPs technical expert groups.

We'd like to provide some explanation to the community as to why we're taking this significant step.

The recent Java SE 7 vote was the last chance for the JCP EC to demonstrate that the EC has any intent to defend the JCP as an open specification process, and demonstrate that the letter and spirit of the law matter. To sum up the issues at stake in the vote, we believe that while continuing to fail to uphold their responsibilities under the JSPA, Oracle provided the EC with a Java SE 7 specification request and license that are self-contradictory, severely restrict distribution of independent implementations of the spec, and most importantly, prohibit the

distribution of independent open source implementations of the spec. Oracle has refused to answer any reasonable and responsible questions from the EC regarding these problems.

In the phrase "fail to uphold their responsibilities under the JSPA", we are referring to Oracle's refusal to provide the ASF's Harmony project with a TCK license for Java SE that complies with Oracle's obligations under the JSPA as well as public promises made to the Java community by officers of Sun Microsystems (recently acquired by Oracle.) This breach of the JSPA was begun by Sun Microsystems in August of 2006 and is a policy that Oracle explicitly continues today. For more information on this dispute, see our open letter to Sun Microsystems.

This vote was the only real power the Executive Committee has as the governing body of the Java specification ecosystem, and as we indicated previously we were looking for the EC to protect the rights of implementers to the degree they are able, as well as preserve the integrity of the JCP licensing structure by ensuring that JCP specifications are able to be freely implemented and distributed. We don't believe this is an unreasonable position—it should be noted that the majority of the EC members, including Oracle, have publicly stated that restrictions on distribution such as those found in the Java SE 7 license have no place in the JCP—and two distinguished individual members of the EC, Doug Lea and Tim Peierls, both have resigned in protest over the same issue.

By approving Java SE 7, the EC has failed on both counts : the members of the EC refused to stand up

for the rights of implementers, and by accepting Oracle's TCK license terms for Java SE 7, they let the integrity of the JCP's licensing structure be broken.

The Apache Software Foundation concludes that that JCP is not an open specification process—that Java specifications are proprietary technology that must be licensed directly from the spec lead under whatever terms the spec lead chooses; that the commercial concerns of a single entity, Oracle, will continue to seriously interfere with and bias the transparent governance of the ecosystem; that it is impossible to distribute independent implementations of JSRs under open source licenses such that users are protected from IP litigation by expert group members or the spec lead; and finally, the EC is unwilling or unable to assert the basic power of their role in the JCP governance process.

In short, the EC and the Java Community Process are neither.

To that end, our representative has informed the JCP's Program Management Office of our resignation, effective immediately. As such, the ASF is removing all official representatives from any and all JSRs. In addition, we will refuse any renewal of our JCP membership and, of course, our EC position.

# # #

Posted at 06:04PM Dec 09, 2010 by Sally in General | Comments[107]

https://blogs.apache.org/foundation/date/20101209

[439] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 11, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[588] where you had referred to parts of the announcements as crap and you said you weren't saying it about the phone, and I said actually do you remember making negative comments about the Android phone; right? And you said you didn't remember.

Now you remember; right?

A. Yes.

Q. All right.

So let's move 5316 in.

THE COURT: Any objection?

MR. VAN NEST: No objection, Your Honor.

THE COURT: Received in evidence.

(Trial Exhibit 5316 received in evidence)

BY MR. BICKS:

Q. And if we can go to the first paragraph there, this is something you wrote on May 2nd, 2009; right?

A. Yes.

THE COURT: Not May 2nd.

THE WITNESS: February 2nd.

MR. BICKS: February 2nd. Thank you, Your Honor.

Q. February 2nd, 2009; right? So this was after your blog; right?

A. Right.

Q. And you say here, "And to this day, even with a horrible product, it's Apple's iPhone versus Google's Android, even though the latter is lame"—those were your words; right?

[589] A. This was Android in 2009, and back when they first introduced it, it was lame.

MR. VAN NEST: Move to strike, Your Honor.

THE COURT: All right. Go ahead.

BY MR. BICKS:

Q. So it—I mean, I want to come back to your blog because you internally said parts of the Google an-

nouncement were crap. You said they were playing fast and loose with the licensing rules. You had no clue. And now you're saying the phone you knew at the time was lame, but none of that information was in your blog; right?

A. The point of the blog was to become a part of the conversation. You know, I had all kinds of internal discussions and dialogues, private conversations. That's not what the blog was for.

Q. But in these private conversations, this was truthful, wasn't it?

A. Yes.

Q. And your statement that they were playing fast and loose with the licensing rules and you had no idea what they were up to, that was truthful?

THE COURT: Did it say licensing rules? I don't think that's the term that was used.

BY MR. BICKS:

Q. Playing fast and loose with licensing terms.

* * *

[924] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 13, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1042] Q. And what does "compatibility" mean?

A. So "compatibility" means that a computer programmer can reasonably expect that their program will behave the same way regardless of which compatible Java implementation they run the program on.

Q. And Sun often referred to that as "write once, run anywhere"; correct?

A. That was in our aspirational goal, yes.

Q. And if you also look—now I'm still on page 9 of 24.

A. Okay.

Q. You see where it says "Do you think anyone will fork the JDK?"

A. Yes.

Q. Can you explain to the jury what a "fork" is?

A. So when this document was written, "fork" meant that a group of software developers would take some open source code and would develop it independently of the original developers of the code, rather than collaborating with them over the same version. And so the result would be that there would be two versions that were not kept in sync with each other.

Q. And this document, this FAQ that Sun prepared, it expressly states in this Q and A that: (reading)

> "Broad distribution of incompatible forks is potentially a danger since such forks could damage the 'write once, run anywhere' compatibility value of the Java [1043] platform."

Correct?

A. That's correct.

Q. And it also says that: (reading)

> "Again, the Java technology compatibility promise is so central to the value of the platform that you would want to protect that."

Correct?

A. Yes.

Q. One more thing on this page, Mr. Phipps. This document also says that by offering OpenJDK under the GPL, that doing so makes forks, proprietary forks—that would be incompatible forks—less likely, does it not?

A. Could you direct me to that? It's quite—that's, again, quite a long paragraph.

Q. It's about—halfway down the question is "So what about compatibility," and it's the fist indent under the answer.

A. Okay. (Witness examines document.)

Q. It says, "License. GPL makes proprietary forks less likely."

A. All right. Okay. Yes.

Q. That's true, isn't it?

A. So that's what the document says, yes.

Q. I'd like to turn to the end of the document. This is at page 15 of 24.

[1044] A. Okay.

Q. And I'm looking at the bottom of the page: (reading)

> "How can Sun have other licenses that bear on the open source code base? Isn't that no longer open source?"

Do you see that?

A. I do.

Q. Can you read what Sun's response was to that question?

A. Okay. The answer says: (reading)

> "Because Sun owns the copyright for the open source code base, Sun is able to license each copy of this code base distributed by Sun under any license, including a commercial software license. This right is inherent in copyright law. Several free and open source communities exhibit this behavior."

Q. And as chief open source officer at Sun, Mr. Phipps, did you talk regularly to open source individuals at Google?

A. It was very infrequent.

Q. Did you speak with Chris DiBona, the director of open source at Google?

A. Around about the time we're talking about here, I did not frequently speak to Chris, no.

Q. Did you speak to him later in your job duties, like in 2010?

A. Yes. Later on in the—so I typically met Chris at community events so—particularly at the open source

* * *

[1047] A. It would be very difficult for me to answer that question.

MS. SIMPSON: I'd like to show the witness trial Exhibit 9191. This has already been admitted.

THE COURT: What part do you want him to look at?

BY MS. SIMPSON

Q. This is an email that you received between you and Jonathan Schwartz; correct?

A. It is.

Q. And if you turn to the second page of the document, it's actually an email that is being sent from Geir Magnusson to Jonathan Schwartz. You testified earlier that Geir Magnusson was the officer at Apache Software Foundation; correct?

A. That—his role was to represent Apache Java in relation to Java matters, yes.

Q. And if you look on the back of the second page of that document, at the very bottom. Do you see where I am?

(reading)

> "Through Apache Harmony, the ASF," that's the Apache Software Foundation, "entered into the specification license in good faith with the expectation that Sun, as the spec lead, would reciprocate."

Do you see that?

A. I do see that.

Q. So Geir Magnusson is indicating that Apache took a specification license, is he not?

[1048] THE COURT: Well, wait. You said that so fast. Say that again but more slowly.

MS. SIMPSON: Sorry.

Q. Geir Magnusson here is indicating that Apache had a specification license for the Harmony project, is he not?

A. That is what it appears to say.

Q. And I'd also like to look at a document Trial Exhibit 2207. It should be up in your folder, Mr. Phipps.

A. Okay. Yes, indeed.

Q. Did you find that?

A. I did.

Q. Can you take a look at it, sir? Is that an email chain between you and Jonathan Schwartz?

A. It is, yes.

Q. And it also contains email from Geir Magnusson?

A. (Witness examines document.) Yes. Yes, it is.

MS. SIMPSON: Move to admit this into evidence,

Your Honor.

THE COURT: Any objection?

MR. KWON: No objection.

THE COURT: All right. Thank you. It's in, but I don't have the number.

MS. SIMPSON: 2207.

THE COURT: 2207 in evidence.

(Trial Exhibit 2207 received in evidence)

[1049] BY MS. SIMPSON:

Q. And this is dated April 10th, 2007; correct?

A. It is.

Q. And if you turn to the second page of the document, Mr. Phipps, where it begins with Geir Magnusson writing to Jonathan Schwartz, do you see the second paragraph that begins "Since August 2006"?

A. Yes.

Q. It says: (reading)

> "The ASF," the Apache Software Foundation, "has been attempting to secure an acceptable license from Sun for the test kit for Java SE. This test kit called the Java Compatibility Kit or JCK"—

Folks also refer to that as the TCK; right, Mr. Phipps?

A. I believe they do, yes.

Q. (reading)

> —"is needed by the Apache Harmony project to demonstrate its compatibility with the Java SE specification as required by Sun's specification license."

Do you see that?

A. I do.

Q. Did you understand that the Apache Software Foundation needed a TCK in order to implement the specification?

MR. KWON: Objection. Calls for a legal conclusion.

THE COURT: Well, if back then you had a view on that

* * *

[1129] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 16, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1272] So in using these API declarations, the Android platform was one that developers could use much more easily. It would have been cumbersome if they hadn't used these same API declarations.

Q. And so you thought it was a sound business practice for Google to leverage the existing community of developers, minimizing the amount of new material and maximizing existing knowledge; isn't that true?

A. That sounds like a reasonable statement. And I'm confident it's something that I said.

Q. All right. And you also agree, sir, that android would not work without the Java APIs that Google copied; isn't that right?

A. Android wouldn't work if you took one line of code out of Android. It would stop working.

So if you took the 37 API package declarations that Google used in creating it, it would also not work. But if you removed the libraries that I spoke about, it also would stop working.

Q. All right. Now, it's true, sir, that you are not an economist?

A. That is true. I am not an economist.

Q. And you did not apply any economic expertise in evaluating the effect of Android on Java SE; isn't that right?

A. That is true.

[1273] Q. And, sir, is it also true that Java ME is a subset of Java SE?

A. There are editions to the Java ME that aren't part of Java SE because it runs on feature phones and embedded devices. But, in general, it's a subset.

Q. Is it true that most of Java ME is a subset of something in Java SE?

A. I think that's reasonable.

Q. Now, you were not offering any opinion in your report about whether SE could be used in mobile phones; is that right?

A. I think that I spoke about how Java SE had not been used in a mobile phone, in my report, other than on the Android platform.

Q. Well, and other than Danger and Savaje, that you heard about while you were sitting here this week; right?

A. Yes. But I believe you asked about my report. And I did not know—I didn't hear about Danger and Savaje before I wrote my opening report.

Q. So that's not something Google called to your attention before you wrote your opening report?

A. I think that's reasonably correct, yes.

Q. Is it true, sir, that you have not offered an opinion whether Java SE is something that could be used in phones?

A. I'm trying to understand. You mean all of Java SE? Because as we've been talking about here, we've used API [1274] declarations from 37 packages to create this new context, Android phone.

If you're asking did I offer an opinion about Java SE on phones, I don't think that's in my report. I think that's right.

MS. HURST: Your Honor, permission to read from the March 2016 deposition, page 134, line 24, through 135 line 6.

THE COURT: Any objection?

Go ahead. No objection.

MS. HURST: (As read:)

"Q. Now, is it your opinion in this case that because Java SE was designed with desktops in mind that it could never be used in phones?

"A. I don't think I have offered an opinion about whether SE would be something that could be used in phones. So I don't—but, so I don't think I offered that opinion, no."

BY MS. HURST

Q. Is it true, sir, that you don't have the business expertise to be able to say what a company's assets are?

A. I don't—as I mentioned earlier, I'm not an economist. Nor am I a specialist in business practices, that's right.

Q. And you've never created software in a commercial setting?

A. I have created software that I've sold.

Q. Has that changed since the time of your last deposition in

* * *

[1322] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 17, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1409] license, and our commercial license. We continued paying Sun for Java.

Q. All right. Let's just look—what's the date on this letter? And let's look at that on our timeline.

A. This is June 22nd, 2007.

Q. So that's before the first release of Android?

A. Yes.

Q. And were you knowledgeable about Android at the time?

A. No. At the time that I signed this letter, I had no idea that Android existed as a project.

Q. All right. And what's the status of Harmony now?

A. Harmony is—is discontinued as a project.

Apache Foundation has a term they use called "in the attic," which means no one is working on it anymore. So Apache Harmony is in the attic.

Q. All right. And are you aware of any unlicensed commercial uses of Apache Harmony in the world today?

A. Only one. Only Android.

Q. Since acquiring Sun, has Oracle continued to license the Java platform?

A. Yes. We license the Java platform in the same ways that Sun did.

Q. And other than the spec license, would you briefly describe what those ways are?

A. So beyond the spec license, of course, we still license

* * *

[1810] UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

No. C 10-3561 WHA

ORACLE AMERICA, INC.,

Plaintiff,

vs.

GOOGLE, INC.,

Defendant.

San Francisco, California
Wednesday, May 19, 2016
Before the Honorable William H. Alsup

TRANSCRIPT OF PROCEEDINGS

---

* * *

[1880] Q. And Google's counsel was pointing out that SavaJE was a phone that was out on the market for a short period of time in the 2004-2005 time frame, generally.

A. Yes.

Q. All right. I want to show you an exhibit that the parties have agreed is admissible. Exhibit 5322.

And this is an exhibit from Mr. Miner to Mr. Rubin.

MR. BICKS: And can we highlight, Trudy, "As a case in point."

BY MR. BICKS

Q. And so we're clear, this is October of 2006. Do you see this, Dr. Jaffe?

A. Yes.

Q. It says here, "As a case in point, if we were not doing what we are doing, SavaJE would probably have gotten more funding."

Do you see that?

A. Yes.

Q. I want to come back to that in a moment.

Remember we were talking on direct about presentations Google was making to OEMs, phone makers, and others about Java?

A. Yeah.

Q. And remember on direct we were talking about presentations that were made in the fall of 2006 and in 2007, that time period generally?

[1881] A. Generally, yes.

Q. Yeah. So what does this tell you, if at all, about the impact of Android on SavaJE's efforts?

A. Well, it says SavaJE would probably have gotten more funding. That means more funding from Sun, to further develop it.

And what it—what it suggests to me is that Sun, at the time, was juggling a lot of things and dealing with the fact that Android was being sold to customers in the sense of OEMs and carriers.

Android was being sold, as we saw yesterday, as a solution that has the Java application framework. And that was part of the challenging environment that Sun was evaluating when deciding how much to invest in SavaJE.

MR. BICKS: Thank you very much, Doctor.

THE COURT: May the witness be excused?

MR. VAN NEST: He may, Your Honor.

MR. BICKS: Yes.

THE COURT: Thank you, Dr. Jaffe. You may step down.

THE WITNESS: Thank you.

(Witness excused.)

THE COURT: I am going to let us break a few minutes early. I need to talk with the lawyers about something. So we're going to take our break at this time.

Please remember the admonition. 15 minutes.

* * *