EXHIBIT A

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,)
Plaintiff,)
vs.) No. CV 10-03561 WHA
GOOGLE, INC.,)
Defendant.)

HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY

Videotaped Deposition of BOB LEE,
taken at 110 Fifth Street, Suite 400,
San Francisco, California, commencing
at 9:35 a.m., Wednesday, August 3, 2011,
before Leslie Rockwood, RPR, CSR No. 3462.

PAGES 1 - 82

Q. When did you join Google?
A. 2004, October.
Q. And when did you join the Android team?
A. I don't remember the exact time. It was pretty early on in the project. I think there were like 12 people on the project or something like that. It was after the acquisition, obviously.
Q. You mentioned you were the core library lead. What are the core libraries for Android?
A. So the core libraries are the lower level libraries. Maybe it's easier to talk about what they're not. They're not the Android framework libraries, which are kind of like how you implement applications, and they're not the UI libraries.
What they are are kind of like all the core functionalities. There's the Java interoperability libraries, and then there's other additional libraries like the networking libraries and things like that.
Q. Mr. Lee, have you ever been deposed before?
A. No.
Q. You understand that you're under -- you've taken an oath to tell the truth today?
A. I do.
Q. And you understand that although we're in this informal setting of a conference room, your

testimony today is as if you were in a courtroom in front of a judge and jury?
A. Correct.
Q. Is there any reason you can't testify to -- competently to facts? Any health issues, medications, intoxicants, anything of the like?
A. Nope.
Q. I'll ask you if you don't understand a question that I ask, if you let me know, and I'll try to rephrase it.
Can you do that?
A. I will.
Q. When you first joined the Android team, who did you work for?
A. I think I reported to Steve Horowitz. He was the director of the team at the time. I actually started out working on the framework libraries.
Q. When did you start working on the core libraries?
A. It was probably a couple months afterwards.
Q. What were your responsibilities -- I'm going to start over.
When did you become core library head for Android?
A. Well, from the beginning. Well, when I

switched to the core library team, for most of the time, I was the core library team. So I led myself.
Q. So you were a --
A. Up until --
Q. -- team of one?
A. -- up until, yeah, about, probably I guess around -- let me think here. Late 2008, early 2009, then I started adding a few more people to the team.
Q. In that time in 2007, who worked on the core libraries besides yourself?
A. There was -- so Dan Bornstein, who was the Dalvik lead, and he actually was responsible for most of them before I took them over. There was one other guy that worked on it shortly before me. I actually forget his name. He didn't work on them for very long. And we also had an outside contractor named Noser that helped out.
Q. What did Noser do for the Android core libraries?
A. Oh, various things. Implemented various parts of it, like, for example, we had to reimplement the SSL libraries. We built them on top of OpenSSL. I think, if I recall, it was mostly for performance, and they did most of that.
They implemented a lot of tests to cover the

missing parts, and they wrote a lot of the documentation, actually pretty much almost all of it.
Q. Besides the SSL libraries, what other -- what other packages did Noser work on?
A. Oh, I don't really -- I don't really remember. I mean, it was just various -- various different ones. They were effectively part of the team.
Q. When you say Noser was effectively part of the team, what do you mean?
A. Well, they -- I mean, they did similar work to what I was doing, which is really dependent on what needed to be done. For a large part of the time, I actually had -- they had one person -- one or two people on site that we worked closely with, and those people were kind of -- they actually did a lot of the work, but they also had a team back in Germany that did some of the more real work.
Q. When you say that Noser had one or two people on site, do you mean Google's Mountain View campus?
A. Correct.
Q. I'll ask you to take a look at what's previously been marked as Exhibit 204.
A. Okay.
Q. Could you take a look at Exhibit 204 and tell me what it is?

There are several other ones, but I don't know how much detail you want me to go into.
Q. Would you say that designing APIs is a creative activity?
MR. PURCELL: Object to the form.
THE WITNESS: Yes, absolutely.
Q. BY MR. PETERS: When you are referring to reimplementing them, are you referring to the work that you did with Noser to implement core libraries according to Java APIs?
A. I am.
Q. If I can ask you to turn to the second page of your self-evaluation.
A. Yes.
Q. In the fourth paragraph there, you have a description that begins: "Most recently I set my sights on our class preloading code."
Do you see that?
A. Yes.
Q. All right. With your work on the class preloading code, was it the case before your work that the list of classes that would be preloaded by Android would be manually managed?
A. Yes, that was the case.
Q. And you wrote that you added hooks to Dalvik

so we can record class load and initialization times.
Can you describe generally what that work involved?
A. Yes. So I mean, basically as you would execute apps, it would performance profile class loading, and it would like time both the loading of a class, which just means really like loading an executable code. And it would also independently measure the initialization time of the class. So when you initialize a class, it runs some kind of code.
And then it would -- so it would record that information, and then I wrote another tool that would -- I mean, this was quite a bit of information. It would mine all of this information that was gathered from running popular apps -- and there's dozens of apps -- and it would mine that information and prioritize and figure out which were the most important classes to preload and have like hot and ready to go into memory. So classes that were already loaded whenever an app started.
Q. So the ultimate output of your tool would be a list of classes and -- that should be preloaded and the order in which they should be preloaded?
A. Correct.
MR. PURCELL: Object to the form.
Q. BY MR. PETERS: And your tool calculated the

list of classes to preload and the order in which to preload them based on data collected from running Dalvik machine; is that right?
MR. PURCELL: Object to the form.
THE WITNESS: What was the question again?
Q. BY MR. PETERS: Yes. Your tool calculated the list of classes to preload and the order in which to preload them based on data collected from a running Dalvik virtual machine; is that right?
MR. PURCELL: Same objection.
THE WITNESS: Yes.
Q. BY MR. PETERS: Now, skipping to the third page of your self-evaluation, there's a section that begins "JCP EC."
A. Yes.
Q. And you wrote that you represent Google on the JCP executive committee as an alternate to Josh Bloch.
When did you begin representing Google on the JCP executive committee?
A. I don't recall the exact date. It was probably around the end of 2005, if I had to guess.
Q. Are you on the JCP executive committee today?
A. No, I am not.
Q. When did you leave the executive committee?

A. When I left Google.
Q. That was towards the end of 2010?
A. The beginning.
Q. The beginning.
What were your general responsibilities as a member of the executive committee representing Google?
A. There weren't a lot. You would attend meetings, and those meetings would be about how the Java Community Process is run and various issues in the community process.
Q. How often did you attend JCP executive committee meetings?
A. I don't remember the exact frequency. It was a couple times per year, was the in-person meetings, and then they had conference calls a couple times per year. I think it was two times per year that there were meetings, and there was like one in Europe and one in the U.S.
Q. Was it ordinarily you who attended those meetings on behalf of Google or Josh Bloch or both of --
A. It was both of us.
Q. Were you familiar with the dispute between Sun and Apache that arose between 2005 and 2010?
A. I don't remember the exact date that it started, but yes, I was very familiar with it.

Q. Did you consult the Java docs when doing your work on the API implementations for Android?
A. Yes.
Q. Okay. And where did you obtain those Java docs?
A. They're posted for free on Sun's website.
Q. Okay. So you consulted Sun's website for the API specifications when doing the work for Google?
A. Yes.
Q. Who else did?
A. I'm not sure.
Q. Did the Noser team consult Sun's specification?
MR. PURCELL: Object to the form.
THE WITNESS: I would -- I don't have any specific knowledge of it, but I would assume so.
Q. BY MR. PETERS: Did you observe any copyright notices on the specifications?
A. Yes.
Q. And what did the copyright notices on Sun's API specifications say?
A. I didn't -- I don't recall.
Q. Did you consult with an attorney about implementing Sun's Java APIs?
MR. PURCELL: You can answer that question

"yes" or "no".
THE WITNESS: No.
Q. BY MR. PETERS: Why not?
A. I didn't see a need to. I mean, it wasn't copying the Java docs.
Q. Before the first commercial release of Android in a mobile device, do you know if anyone at Google consulted with an attorney regarding the Java API implementations in Android?
A. I don't have knowledge of that.
Q. Do you know if anyone at Google investigated Sun's patent portfolio before the first commercial release of Android in a mobile device?
A. No.
Q. Did you have discussions on the Android team about Sun's intellectual property rights?
A. No.
Q. Earlier today you mentioned that Dan Bornstein had been working on the Java API implementations before you joined up.
A. Correct.
Q. What resources did Mr. Bornstein do in his work on the Java API implementations?
MR. PURCELL: Object to the form.
THE WITNESS: Actually, I do want to go back

in -- to your previous question. I mean, you mentioned intellectual property. So obviously the trademark is intellectual property. I knew that -- yeah, I did discuss getting that Java stamp of approval.
Q. BY MR. PETERS: So those were discussions about not being able to call things Java, because they had not passed the TCKs; is that right?
A. Correct. Yeah.
Q. What resources did Mr. Bornstein use in his work on implementing the Java API specifications?
MR. PURCELL: Object to the form.
THE WITNESS: I'm not sure.
Q. BY MR. PETERS: Did Mr. Bornstein consult Javadoc, to your knowledge?
MR. PURCELL: Object to the form.
THE WITNESS: I'm not sure.
Q. BY MR. PETERS: You had -- turn back to Apache and Apache Harmony for a minute.
A. To what? I'm sorry.
Q. To Apache Harmony. Not any specific document.
A. Oh, okay.
Q. But you had mentioned that Sun was putting on a field of use -- trying to put on field of use restrictions onto Apache that would prevent the use of

Harmony in a mobile device; is that right?
A. Uh-huh. Yes.
Q. But Google took Harmony to use in a mobile device; isn't that right?
A. Yes. This other stuff came out quite a bit after, though, to my knowledge.
Like, for example, Android used Harmony, and then I don't really recall being exposed to the actual issues with Harmony until sometime later.
Q. Android wasn't released as an -- as a mobile device, a commercial product, until 2008; isn't that right?
A. I don't remember the exact date. That sounds about right.
Q. But -- so if I understand you correctly, Google started using Harmony code before their dispute between Apache and Sun --
A. Oh, long before. Yeah.
Q. When did Google start using Harmony code?
A. I don't remember exactly. I can tell you it was before, like, open JDK and all that stuff was even announced.
Q. Was it, I guess, one of the things that Noser did was to take Harmony code and adapt it for Android?
A. Yes.

I declare under the penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Executed on September 7th, 2011, at _____, _____.

_____

SIGNATURE OF THE WITNESS

STATE OF CALIFORNIA) ss:
COUNTY OF MARIN)

I, LESLIE ROCKWOOD, CSR No. 3462, do hereby certify:

That the foregoing deposition testimony was taken before me at the time and place therein set forth and at which time the witness was administered the oath;

That testimony of the witness and all objections made by counsel at the time of the examination were recorded stenographically by me, and were thereafter transcribed under my direction and supervision, and that the foregoing pages contain a full, true and accurate record of all proceedings and testimony to the best of my skill and ability.

I further certify that I am neither counsel for any party to said action, nor am I related to any party to said action, nor am I in any way interested in the outcome thereof.

IN WITNESS WHEREOF, I have subscribed my name this 4th day of August, 2011.

Leslie Rockwood

LESLIE ROCKWOOD, CSR. NO. 3462

EXHIBIT D

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,)

Plaintiff,)

vs.) No. CV 10-03561 WHA

GOOGLE, INC.,)

Defendant.)

-- HIGHLY CONFIDENTIAL, ATTORNEYS' EYES ONLY--

Videotaped Personal Capacity deposition of ANDREW E. RUBIN, taken at the law offices of King & Spalding LLP, 333 Twin Dolphin Drive, Suite 400, Redwood Shores, California, commencing at 8:39 a.m., on Wednesday, July 27, 2011, before Leslie Rockwood, RPR, CSR No. 3462.

PAGES 1 - 296

So I'm assuming Steve's using JVM in the generic sense to refer to either the contemplated partnership that we're negotiating with Sun or our internal clean room implementation. I don't know which he's referring to. It's hard for me to -- it's hard for me to read into that.

Q. But is it your testimony as of March 23, 2006, you had a -- what you regarded as a viable fallback strategy such that the Sun JVM was in fact not core to your platform architecture and strategy?

MS. ANDERSON: Objection. Form.

THE WITNESS: I believed that I was developing a viable fallback strategy in case we weren't able to reach partnership conclusion with Sun.

Q. BY MR. JACOBS: And you believed that at the time?

A. Yes.

(Exhibit PX314 was marked for identification.)

Q. BY MR. JACOBS: I'd like you to take a look at an email string between you and Greg Stein around March 24, 2006?

A. Yes.

Q. So who was Greg Stein?

A. I don't know. I believe he's somebody that

reported to Chris DiBona in the Open Source product team. Again, that's the team of evangelists that go off and evangelize Open Source to the community.

Q. So he starts out "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 -- and I can't tell them about our effort. IOW."

Do you know what IOW is?

A. I have no idea.

Q. -- "IOW, without violating confidentiality somewhere, there's no way for me to make the two parties aware of each other's 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 info that I can give them so that we can open the door more?"

Do you see that?

A. Yep.

Q. So we get the problem. He's got information on both sides that he can't share with the other about some plans to Open Source J2ME, right?

MS. ANDERSON: Objection. Form.

THE WITNESS: Yes.

Q. BY MR. JACOBS: And so he's asking you if you can -- if you can give little information about your perhaps to the other side and so that he can create a channel of communication, correct?

MS. ANDERSON: Objection. Form.

THE WITNESS: Well, I think it's a little deeper than that. Chris DiBona said I'm the right person to talk to about our plans because I'm not sure Greg knew what our plans were. These are two teams at Google. Greg's not an engineer on the Android engineering team. So I think he somehow was representing an opportunity to talk to somebody, "them," in his sentence, and he needed more information about what we were doing and he's coming to me to get more information.

He was coming to me both to get approval to talk to somebody that I don't know who they were and more information about what I was doing and I responded with a two-sentence response giving him no more information.

Q. BY MR. JACOBS: In which --

A. In my typical, you know, New York style.

Q. Okay. So you say "I don't see how you can open Java without Sun since they owned the brand and I'd be happy to talk."

So you're really saying you're pretty skeptical about this other plan; right?

A. Look, he didn't give me any information about the them, the who, you know, and again, my view on these things is there's all sorts of people at Google who want to know what we're doing and have brilliant ideas and you've seen, you know, some of those people with great bravado, you know, challenging the strategy on email strings.

Generally speaking, these aren't strategy guys, these are engineers, individual contributors who are full of innovative ideas but somehow need to be harnessed in a way that's actually productive for the company.

Q. When you say -- he says "They have a plan, the ability to call it Java(tm) is simply a matter of passing the J2ME TCK as I understand it."

Do you see that?

A. Yes.

Q. And when you write back "Ha, wish them luck. Java.lang API's are copyrighted and Sun gets to say who they license the TCK to and forces you to take the shared part which tanks any clean room implementation."

Do you see that?

A. Yes, I do.

Q. So you were -- let's start with "Java.lang API's are copyrighted."

What did you mean by that?
A. So this is all information that I learned while I was at Danger because I took license from Sun while I was at Danger. So I had knowledge of their practices around offering people licenses and the constructs they used to make sure that they have an ongoing future licensing revenue opportunity.
Q. And in particular, you understood that Sun was relying on copyright protection for the API's, correct?
A. One of -- one of Sun's arguments to me while I was at Danger is that the -- they thought the Java API's were copyrightable.
Q. And therefore?
MS. ANDERSON: Objection. Form.
THE WITNESS: There was no therefore.
Q. BY MR. JACOBS: Therefore you need today take a license?
MS. ANDERSON: Objection. Form.
THE WITNESS: No, I don't believe that.
Q. BY MR. JACOBS: Was the context of the argument while you were at Danger?
A. I think it was a bad argument.
Q. What was the context?
MS. ANDERSON: Objection. Form.

THE WITNESS: I don't recall the exact contents of the article. I don't think it was used as a threat, if that's what you're referring to.
Q. BY MR. JACOBS: And did you at some point form an opinion that this argument was weak?
A. Around copyright?
Q. Yes.
A. Well, I didn't think -- I don't think that, in general, I don't care whose API's. I don't believe that API's are copyrightable.
Q. And where -- when did you form that judgment?
MS. ANDERSON: Objection. Caution the witness to the extent responding would cause you to reveal communications with counsel, I instruct you not to answer on the grounds of attorney-client privilege.
THE WITNESS: Yeah. Generally speaking, it's of my opinion and this, again, we're talking about my personal opinion as a computer scientist, that these API's and API's in general are documented. Sometimes books are written about API's. Sometimes there have been legal cases about whether API's are copyrightable and based on just industry knowledge, I didn't have a belief personally that API's were copyrightable.
Q. BY MR. JACOBS: Did you rely on that belief in ultimately pursuing your course of development on

Android?
MS. ANDERSON: Objection. Form.
THE WITNESS: No, the issue never really came up. Nobody, nobody threatened me in any way around copyrighting API's during the, you know, the four years of negotiations with Sun. Didn't even come up.
Q. BY MR. JACOBS: And did you on your own think of that as an issue as you were making development decisions with respect to Android?
MS. ANDERSON: Objection. Form.
THE WITNESS: Again, I thought it was non-issue. So it didn't even enter my mind that somebody would actually try to apply that to my work.
Q. BY MR. JACOBS: While you were at Danger had heard from Sun that they -- that Sun had -- you used the word here "copyrighted," the Java language API's, so you were aware of that. You put it in an email around -- in an email string that has at least tangential relationship to Android; right?
MS. ANDERSON: Objection. Form.
THE WITNESS: Well, so first of all, I don't know if Sun Java API's are copyrightable, are they?
Q. BY MR. JACOBS: Well, you wrote -- I just want to know what you had in mind when you wrote this email.

A. So again, like I don't really have -- I'm not a lawyer, okay. I didn't hit any books and I didn't learn whether these are copyrighted.
I was relying to acknowledge that I learned at Danger during my negotiations with Sun where I thought there was a possibility or at least they inferred to me that there was a possibility that they thought these were copyrightable.
Q. And at Danger you successfully negotiated with Sun and you took a Sun license, correct?
A. Yes. That's correct.
Q. And in the development of Android, ultimately you proceeded without a Sun license, correct?
MS. ANDERSON: Objection. Form.
THE WITNESS: Ultimately we got to the -- we decided after a long period of negotiations that it was clear Sun didn't want to partner with us on -- and we built a clean room implementation.
Q. BY MR. JACOBS: Ultimately you proceeded in the development of Android without a Sun license, correct?
A. Remember, I was always implementing clean room version as a hedge, so I would reword that to say ultimately I released a clean room implementation of the Java programming language and a virtual machine that

I declare under the penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Executed on AUGUST 31, 2011, at MOUNTAIN VIEW, CA.

SIGNATURE OF THE WITNESS

STATE OF CALIFORNIA) ss:
COUNTY OF MARIN)

I, LESLIE ROCKWOOD, CSR No. 3462, do hereby certify:

That the foregoing deposition testimony was taken before me at the time and place therein set forth and at which time the witness was administered the oath;

That testimony of the witness and all objections made by counsel at the time of the examination were recorded stenographically by me, and were thereafter transcribed under my direction and supervision, and that the foregoing pages contain a full, true and accurate record of all proceedings and testimony to the best of my skill and ability.

I further certify that I am neither counsel for any party to said action, nor am I related to any party to said action, nor am I in any way interested in the outcome thereof.

IN WITNESS WHEREOF, I have subscribed my name this 28th day of July, 2011.

Leslie Rockwood

LESLIE ROCKWOOD, CSR. NO. 3462

EXHIBIT E

From:	Bill Coughran.	Sent:2/10/2006 6:30 PM.
To: [-]	lindholm@google.com.	
Cc: [-]	arubin@google.com.	
Bcc: [-]	.	
Subject:	Re: Travel for Android requested.	

OK

Sent from my BlackBerry Wireless Device

-----Original Message-----
From: Tim Lindholm <lindholm@google.com>
To: Bill Coughran <wmc@google.com>
CC: Andy Rubin <arubin@google.com>
Sent: Fri Feb 10 18:17:29 2006
Subject: Travel for Android requested

Hi Bill,

As you might vaguely be aware, I have been helping Andy Rubin with some issues associated with his Android platform. This has mostly taken the form of helping negotiate with my old team at Sun for a critical license.

Andy had also asked Frank Yellin and I to get involved in the evaluation of a Java acceleration architecture (silicon and surrounding software) done by TI. This work has been going on for years. TI is now trying to figure out whether it has value and, if so, how to bring it to market. They would really like to team up with Google/Android, and preferably Sun, to go out in future Google Android handsets. TI has already come in and proposed their architecture to the Android group. I crashed that meeting, and all agreed it sounds promising, but Andy doesn't think his team has the right background to evaluate the work more fully.

This has now led to a request from TI for Frank and I to do a two-day deep dive into the architecture with TI's engineers. The proposal is to do it at their place because that's where all their engineers are, and their place is in France.

The proposed date is the end of the month, and Frank has got an OK from his management already. Would you be supportive of me doing this? It would have to be a quick trip given the other things going on.

Thanks,

-- Tim

EXHIBIT F

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,)

Plaintiff,)

vs.) No. CV 10-03561

GOOGLE, INC.,)

Defendant.)

HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY

Videotaped Deposition of TIM LINDHOLM, taken at 333 Twin Dolphin Drive, Redwood Shores, California, commencing at 9:56 a.m., Wednesday, September 7, 2011, before Ashley Soevyn, CSR No. 12019.

PAGES 1 - 115

has a personal counsel as well.
THE WITNESS: Should I repeat that?
MR. NORTON: I would like the witness to answer the questions.
THE WITNESS: Okay, yes, Ms. Anderson is my -- is my -- is counsel for Google, and Michael is my personal lawyer.
MR. NORTON: Thank you.
Q. And is Google paying for your personal lawyer?
MS. ANDERSON: Objection to the extent responding to this question would require you to reveal information you only know through communications with counsel, I instruct you not to answer on the grounds of privilege. Otherwise, you may answer.
THE WITNESS: I only know anything about that through communication with my attorneys.
MR. NORTON: I don't think that's a proper instruction objection, and I'm not going to take up limited time with that today.
Q. Mr. Lindholm, you began working at Sun Microsystems in March 2004; is that right?
A. March 200- at Sun Microsystems? No, I think it was 1994.

Q. I beg your pardon, 1994. Thank you.
A. And March sounds approximately correct. It was a long time ago, but I'm not sure that I'm accurate about that.
Q. But the year was 1994; is that right?
A. I think that's correct.
Q. And you continued to work at Sun until what year?
A. I believe I went from Sun to Google in 2005.
Q. Around July of 2005; is that correct?
A. That's my recollection yes.
Q. And during the time that you were at Sun, did you work on Java?
MS. ANDERSON: Objection, form.
MR. LISI: Join.
THE WITNESS: During the time I was -- well, so Java consisted of many things. There were parts of Java -- parts of the Java technologies that I did work on during that time. I did not work on all aspects of Java, just selected ones.
BY MR. NORTON:
Q. Can you briefly summarize for me the aspects of Java on which you worked during ten years you were employed at Sun Microsystems?

MS. ANDERSON: Objection, form.
THE WITNESS: It's a fairly long list. But to start with, I initially worked on -- in the original Java team, which was involving the original creation of the Java platform. I continued working on that for a number of years, and then at some point, switched over as Java -- as the Java plat- --the original technology was broken up into various subplatforms that are today known as the editions, typically. I began to work more on the edition being used for mobile and embedded software.
BY MR. NORTON:
Q. And is there a name for the edition for mobile embedded software?
A. Yes, typically, Sun would have -- Sun and Oracle and the public would know this as Java Micro Edition.
Q. It was called Java ME, or Java 2 ME?
A. Yeah, it's gone through various -- it went through various abbreviations over time and the "2" was eventually removed. So I think today, as far as I know, it's Java -- or JME.
Q. One of the aspects of Java that you worked on was the Java Virtual Machine; is that correct?
A. That's correct.

Q. What is the Java Virtual Machine?
MS. ANDERSON: Objection, form.
THE WITNESS: Well, so, first off, it's a virtual machine, you wouldn't be surprised by. It's the virtual machine that was used by Sun to run the Java programming language on. I'm not sure how deeply you would like me to go into this.
BY MR. NORTON:
Q. Well, can you describe for me, in brief form, in two sentences, what, in essence, does the Java Virtual Machine do?
MS. ANDERSON: Objection, form.
THE WITNESS: Well, I would rather say I would rather start with saying what a virtual machine does.
BY MR. NORTON:
Q. That's fine. Why don't you do that first?
A. Okay. A virtual machine is is a general concept in computer science in many virtual machines that existed over time. It's typically I don't know if there is a crisp, formal definition that people in the field would uniformly agree is the valid one, but in a general sense, a virtual machine is typically described as an abstract computing

machine that will typically include some sort of instruction set that is intended to be -- well, is typically implemented as a software program.

This is -- I'm meaning to draw the distinction with a concrete machine, a concrete processor, such as, say, an Intel chip. An Intel central processing unit chip will, itself, have a set of instructions, but those instructions and the other aspects of the design of that chip are intended to be -- are typically intended to be implemented in silicon --

THE REPORTER: In what?

THE WITNESS: In silicon as hardware.

So a lot of -- there is a bit of a gray area here in that it would be perfectly reasonable for me to do a software implementation of an Intel instruction set and to emulate the Intel instruction set. In fact, various -- various companies have done that sort of thing is my general understanding.

For instance, when Apple transitioned from the Power PC platform to the Intel platform, I think that they did an emulator for the Power PC platform so that people who had old applications that were compiled to the Power PC, could be run on the now Intel-based, new Macintoshes.

So to get back to the start, a virtual machine is an abstract machine typically meant to be implemented in software versus in silicon or hardware, like a processing processor specification that is in silicon.

BY MR. NORTON:

Q. And you wanted to describe a virtual machine generally, and then my question to you previously had been could you tell me in two sentences what a Java Virtual Machine is?

MS. ANDERSON: Objection, form.

THE WITNESS: Well, so -- the -- there have been a number of -- there have been a number versions of the Java Virtual Machine over time. But generally speaking, a Java Virtual Machine refers to a particular virtual machine architecture that was specified by Sun and implemented -- has been implemented by Sun and by others. But it's a particular virtual machine design that has particular features and particular properties.

BY MR. NORTON:

Q. And were you one of the contributors to the original Java Virtual Machine?

MS. ANDERSON: Objection, form.

MR. LISI: Join.

THE WITNESS: When I joined Sun in 1994, Java didn't exist. There was no language by -- no language, no virtual machine named J-a-v-a, or JVM. The -- as is become public now, at that time -- at that time Sun was working on a language that's called Oak, and Oak had a virtual machine.

Over time, beginning shortly after I joined, things happened in the project that had led to the creation of Oak such that -- such that the Oak project was basically canceled. But some of the technology that was developed in the context of Oak was repurposed for the World Wide Web and was renamed Java.

So the Oak virtual machine, which predated my presence at Sun, contained a large portion or probably, arguably the majority of the Java Virtual Machine design. But over the next series of years -- and actually, continuing to the current day, the Java Virtual Machine had continued to evolve over time. But in particular, during the time not long after I joined Sun, the Java Virtual Machine was firmed up, fleshed out --

Q. Mr. Lindholm, I don't like to interrupt the witness, but -- and I appreciate your -- you're extremely knowledgeable about this, but I would like

to focus on my questions particularly --

A. Okay.

Q. -- since we do have a very limited amount of time.

A. Okay, I'm sorry.

Q. And my question was just a "yes" or "no" for now --

A. Okay.

Q. -- which was, were you a contributor for the original Java machine -- Virtual Machine, I should say?

MS. ANDERSON: Objection, form.

MR. LISI: Join.

THE WITNESS: Well, so I can't give a crisp "yes" or "no" in the context of what I was saying. I was a contributor to the early, not the earliest, but in the early days of Java, and the early Java being brought to the public eye, I was a contributor at that point.

BY MR. NORTON:

Q. And you are the coauthor of the book, "The Java Virtual Machine Specifications"; is that correct?

A. That's correct.

Q. And what is the purpose for which the Java

Virtual Machine specification is used?
MS. ANDERSON: Objection, form.
MR. LISI: Join.
THE WITNESS: Well, it's used for a number of purposes. It it was intended to be it was intended to be the definitive statement of what the Java Virtual Machine definitive definition of the Java Virtual Machine. It was also used by people who wanted to implement Java Virtual Machines or who wanted to certainly, in the early days, people were just very interested in Java technology. And a lot of people wanted to read the virtual machine specification just to understand how Java technology worked. So it had quite a number of intents, and I'm sure people use it for other purposes, too.
Q. Your coauthor on the Java Virtual Machine Specification was Frank Yellin; is that correct?
A. That's correct.
Q. And Mr. Yellin is also employed by Google; is that correct?
A. As far as I know, he was when I last talked to him.
Q. You were a key contributor to the Java programming language, correct?
MS. ANDERSON: Objection, form.

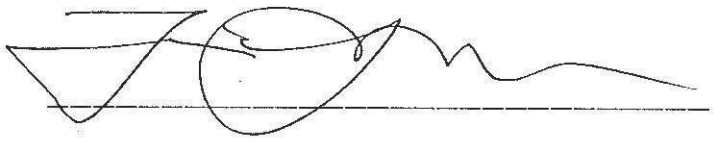
MR. LISI: Join.
THE WITNESS: I don't think I would call myself a key contributor to the Java programming language. I contributed some things, but there were certainly quite a few people who did much more than that I did. I was more focused on the virtual machine.
BY MR. NORTON:
Q. You were a key contributor to the Java Runtime, correct?
MS. ANDERSON: Objection, form.
MR. LISI: Join.
THE WITNESS: The virtual machine is part of the Java Runtime, so -- so at least by that, yes.
BY MR. NORTON:
Q. And you were an original member of the Java technology project at Sun, correct?
A. Well, yes, I'm not sure if in some official sense there was a Java technology project. Various people used various things to say that they were there at the beginning when Java, as it became known to the public, what was done. So, you know, I -- what that says is I was present -- I was in the team in '94, '95.

MR. NORTON: Let's mark our first exhibit of the day, which will be Plaintiff's Exhibit 524. I will hand that to you, Mr. Lindholm. I have copies for counsel.
(Exhibit 524 marked for identification.)
MS. ANDERSON: Thank you.
BY MR. NORTON:
Q. Mr. Lindholm, have you ever seen that before?
A. Yes, I have.
Q. Is that a website biography of you?
A. I believe that it is.
Q. Did you write it?
A. I think that originally I did not write it. I might have commented on it, but -- well, I do not believe I originally wrote it.
Q. But you did comment on it?
A. I had seen it, yes.
Q. Did you see it before it was published, that is, posted on the Internet?
A. I don't remember. This was -- this was originally written quite some time ago.
Q. And have you seen it on the Internet?
A. Yes, I think I have.
Q. Did you ask anybody to change anything in

the website bio in Exhibit 524?
A. I don't remember. I don't recall. I think that -- I think that there were -- I think that in the past, whether it was -- I think that in the past I have asked people to change -- I think I have asked something to be changed in here, but I don't really remember what or what the context of that was.
Q. You'll see that in the second paragraph of the biography it states, "Prior to Google, Tim was an original member of the Java technology project at Sun Microsystems and a key contributor to the Java programming language and Runtime, both definition and implementation." Do you see that?
A. I do see that.
Q. That was part of the bio that you'd seen before?
MS. ANDERSON: Objection, form.
THE WITNESS: I don't know. This wasn't something that I've spent a lot of time worrying about. This is a small -- this was something that somebody else wanted to put up about me, and I said -- kind of said, "Okay."
BY MR. NORTON:
Q. You haven't changed it, have you?

I declare under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

Executed on October 18, 2011, at Mountain View, California.



TIM LINDHOLM

STATE OF CALIFORNIA) ss:
COUNTY OF MARIN)

I, ASHLEY SOEVYN, CSR No. 12019, do hereby certify:

That the foregoing deposition testimony was taken before me at the time and place therein set forth and at which time the witness was administered the oath;

That the testimony of the witness and all objections made by counsel at the time of the examination were recorded stenographically by me, and were thereafter transcribed under my direction and supervision, and that the foregoing pages contain a full, true and accurate record of all proceedings and testimony to the best of my skill and ability.

I further certify that I am neither counsel for any party to said action, nor am I related to any party to said action, nor am I in any way interested in the outcome thereof.

IN THE WITNESS WHEREOF, I have transcribed my name this 8th day of September, 2011.

ASHLEY SOEVYN, CSR 12019

EXHIBIT G

From:	Andy Rubin.	Sent:8/9/2005 4:28 PM.
To: [-]	Tim Lindholm.	
Cc: [-]	.	
Bcc: [-]	.	
Subject:	Re: Project advisors.	

Sorry --

The three names I mentioned were for the Java advisors, with you being the charter member ;-)

UI is Scott Jenson
and Browser are the key players of the firefox team.

I agree with you, the scars and specific knowledge you have of the wireless ecosystem is key to helping us do the right thing. The others exist on a different plane, and I don't expect them to be involved day to day.

Thanks again for your help.

- a

On Aug 9, 2005, at 3:25 PM, Tim Lindholm wrote:

> Hi Andy,
>
> Sure, sign me up. I think that this only puts a name on what I've
> already been doing and hope to keep doing.
>
> I don't know Patrik, but between me, Robert and Urs I don't think we
> know much of anything about UI or browsers. Robert knows a lot about
> the guts of a small JVM implementation; Urs maybe not so much -- he
> was involved in the original big HotSpot, but not its transition to
> little things. He's a smart guy and a good emeritus advisor if you
> can get him, of course. I think my main value would be as J2ME
> runtime generalist and interpreter of the engineering/business/legal
> ecosystem. Outside of your own team I don't know of anybody else at
> GOOG that can help much with that.
>
> -- Tim
>
> Andy Rubin wrote:
>
>> Tim,
>> So I'm instituting an idea called Project Advisors for Android.
>> The areas that the team could benefit from are:
>> - Java
>> - UI
>> - Browsers
>> I was wondering if I could add your name to the Java advisor
>> team. You're the first I'm asking. I'll probably also ask
>> Patrik Reali, Robert Griesemer and perhaps Urs Hoelzle.
>> What are your obligations as an advisor? None, especially since
>> you have already been part of our extended team.
>> We'd simply just add your name on the advisor page of our wiki in
>> recognition of your contributions.
>> Let me know if this is okay.
>> - andy
>

EXHIBIT H

Contact Report

Leo Cizel , Oracle Corp.
475 Sansome St.
15th Floor
San Francisc , CA 94111

Contact

Company	Google, Inc.	*Address 1*	1600 Amphitheatre Pkwy		
Contact	Andy Rubin	*Address 2*			
Title	VP, Android Project	*Address 3*			
Phone	[redacted] *Ext.*	*City*	Mountain View		
Fax		*State*	CA	*Zip*	94043
Salutation	Andy	*Country*			

User Fields

User 1		*User 9*	
User 2		*User 10*	
User 3		*User 11*	
User 4		*User 12*	
User 5		*User 13*	
User 6		*User 14*	
User 7		*User 15*	
User 8			

Home/Phone

Alt Phone	*Ext.*	*Home Address 1*			
Mobile Phone	[redacted]	*Home Address 2*			
Pager		*Home City*			
Home Phone		*Home State*		*Home Zip*	
E-mail Address	arubin@google.com	*Home Country*			

Alt. Contacts

Assistant	Tracey Cole (tcole@...)	*3rd Contact*	Dan Boornstein		
Asst. Title	Admin. Assist.	*3rd Title*	VM Architec:		
2nd Contact	Brian Swetland	*Spouse*			
Asst. Phone	[redacted] *Ext.*	*3rd Phone*		*Ext.*	
2nd Title	Lead VM Architect	*Referred By*			
2nd Phone	*Ext.*	*Web Site*			

Status

Last Reach		*ID/Status*	
Last Meeting		*Last Results*	
Last Attempt		*Public/Private*	Public

Notes/History *Date Range:* All Dates

Note	11/12/07	5:44 PM	Google today released the Android S/W SDK
Note	5/26/06	7:53 PM	After many meetings incl. Alan Brenner, it was agreed that the two companies cannot come to a meeting of minds on how to work together re CDC-HI and open source.
Note	8/19/05	7:52 PM	Andy is interested in CDC-HI for wireless devices.
Note	8/5/05	10:45 PM	Andy informed me today that the co. by which Android Research was acquired was Coogle. Vineet and I w/meet w/Andy and Tim Lindholm at on Fri. 8/19 at Google.

OAGOOGLE0100029446

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
TRIAL EXHIBIT 2720
CASE NO. 10-03561 WHA
DATE ENTERED _____
BY _____
DEPUTY CLERK

EXHIBIT I

From:	Andy Rubin.	Sent:4/20/2006 1:36 PM.
To: [-]	Rich Miner.	
Cc: [-]	.	
Bcc: [-]	.	
Subject:	Re: prd.	

Enclosed.

Still working to incorporate our conversation...

Android
Product Requirements

Google Proprietary and Confidential

Table of Contents

Introduction

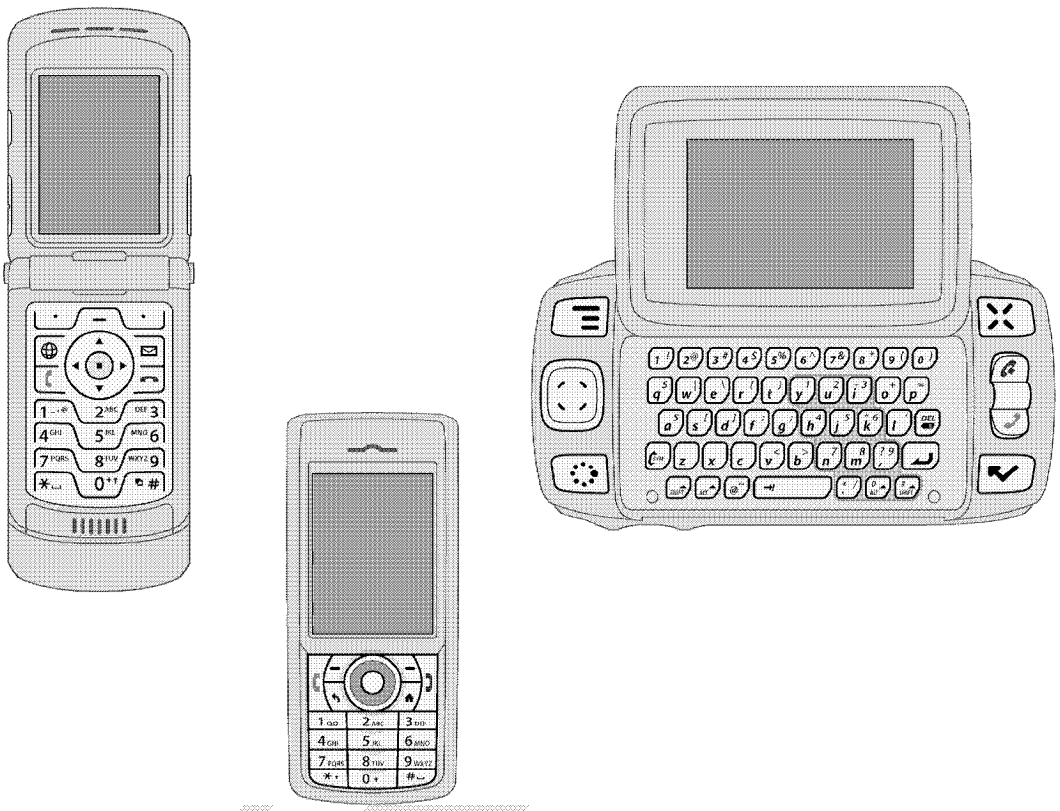
Google's goals for the Android experience can be summarized as follows

- First and foremost, phones using the Android platform will be better phones. Designed with simplicity in mind, these phones should be easier to learn and easier to use.
- These phones will also optionally provide great interfaces to Google services and applications, as well as the means to easily and smoothly add 3rd party applications.
- Finally, Android phones will provide a rich media experience competitive with emerging consumer phones, including mp3 and video capabilities.

1 Vision Statement

A message from Andy". TBD.

2 Hardware Requirements



Minimal hardware requirements

RAM: 64MB
Non-volatile internal storage (ROM): 64MB
Non-volatile external storage (expansion slot): Mini SD
CPU: ARM 9 200MHz
Baseband technology: GSM EGPRS
Primary Display: QVGA TFT LCD, 16-bit color
Secondary Display: Optional (see section 3.2)
Bluetooth 1.2
Notification: Vibration, RGB message LED
Keypad: 0-9, *, #
Navigation: 5-way D-pad, Home, Back, two soft menu keys
Call control: Call, End, Power, Volume up/down
Camera: 2MP CMOS, dedicated picture button
Removable Li-ion battery
USB interface
Speakerphone, full duplex
Headset: 2.5mm stereo headset jack
wifi support

Optional hardware features

Secondary display
Baseband technology: UMTS/HSDPA, CDMA/EvDO
Bluetooth 2.0 EDR
3D Graphics Acceleration
IrDA
GPS: Assisted GPS preferred
QWERTY keyboard

3 System Software

Blah, Blah, Words, Words.

Platform Infrastructure

Platform Infrastructure includes the necessary low level firmware for booting the device form a clean state, recovering the device if firmware has become disabled, updating the various system components, such as baseband firmware, boot firmware and operating firmware. The system will have support for updating both wired and wirelessly, “over the air” (see section 7.2.2). Finally, the platform will include functionality for error reporting and logging, which will be centrally collected and analyzed to ensure quality and prioritization of future updates.

Boot Loader

The Android Boot loader is responsible for taking control of the CPU on initial power up (or power up from deep sleep mode), initializing the CPU and core peripherals, and transferring control to the Linux kernel. The boot loader also provides a USB debug/recovery interface that can be used to boot the device from USB for hardware bring-up, initial installation, or upgrade/reinstall of 'bricked' systems. The boot loader will be 10-40k of code, depending on specifics of the Flash part used, complexity of the core system, etc.

Recovery

Recovery mode is a feature of the Boot loader and/or a minimal kernel + recovery program environment used to reinstall system software in the event of a flash programming failure or other catastrophic event resulting in system corruption. The intent of the recovery system is that it will never need to be rewritten (and possibly it will be physically impossible to do so) once the device in factory programmed, providing a safety net that will always be there.

Remote Debugging Interface

A USB based remote debug/diagnostic interface is provided over the extensible APROTO (Android Protocol). Currently this bridge allows for sending applications and data to the device, inspecting device state using a remote shell interface, and observing kernel debug messages. It is also used to provide transport for remote GDB (native C/C++) debugging and remote Java debugging. Future versions will provide more state inspection tools and more transparent filesystem bridging (so that the device can mount local filesystems from the debugging/development host machine).

Logging

The device needs to provide debug logs for developers and event logs for service providers. Debug logs are verbose text messages that can be used to examine system state while developing software. Event logs are an application feature, described later. The debug logs need to work from multiple processes on the device, simulator, and emulator. The logging system allows log messages to have priority levels (e.g. error, warning, info, debug) and a “tag” that identifies the message source. This way, uninteresting messages can be filtered out without

having to recompile the code. Advanced log viewer apps can retain all log messages and dynamically re-filter past output, which can be helpful when debugging problems that are difficult to reproduce.

Application Framework

The Application Framework is the primary interface by which a third party developer creates applications and services that extend the platform. It is meant to provide a clean set of APIs that can be used equally by internal developers extending the system software, as well as external developers building and extending the platform in unforeseen ways. Therefore, it is one of the most complex and important functions of the system.

Common Navigation

The Application Framework will allow applications written by both internal and external developers to be combined into a common experience. For example, the phone may have a contacts application, a mail application and a maps application written by three different companies, and the user will be able to navigate both forward and backward through these applications without being aware of application or process boundaries.

Activities and Intents

The Android UI will be modular and allow individual applications and components to be replaced. This is accomplished by using intents and activities. An intent is a data structure that is a formal representation of something the user has asked do to (e.g. choose a picture, or make a call to someone). Intents are bound at runtime to activities, which are the objects that implement the UI for performing those tasks. Behavior can be modified, extended or replaced by changing the mapping from intents to activities.

Content Providers

Content providers create an abstraction layer between the physical representation of data and clients who consume that data. This allows components to be swapped both above and below the abstraction. It is possible to create multiple clients that consume the same data, and it is possible to change the data source without requiring changes to the client application.

Messaging (IAC)

Application Manager

Responsible for starting, stopping, pausing applications, and notifying other applications of events.

View System

Compositing

Text Subsystem

The text package is intended to provide the system and applications with all the tools they need to display and to enable users to edit multilingual styled text. It contains storage classes that allow text to be manipulated and to have styles attached to ranges of it; a layout class that arranges the text into lines and interfaces with the graphics system to draw the text; a dynamic layout class that updates itself when the text or its styles change; and input methods that allow the user to input, modify, or select text using the keypad and buttons.

Text Input

Dictionary based.

FreeType

FreeType 2 is a software font engine that is designed to be small, efficient, highly customizable, and portable while capable of producing high-quality output (glyph images). It can be used in

graphics libraries, display servers, font conversion tools, text image generation tools, and many other products as well.

Note that FreeType 2 is a font service and doesn't provide APIs to perform higher-level features like text layout or graphics processing (e.g., colored text rendering, 'hollowing', etc.). However, it greatly simplifies these tasks by providing a simple, easy to use, and uniform interface to access the content of font files.

Control Framework

Widgets

XML Layout

Graphics Plugins

Externally defined graphic formats, with possible animation and interactivity will be supported through a plugin architecture, allowing them to be hosted inside any application, and as content inside a web page. Examples of note are Flash and SVG.

Graphics Subsystem

An application API for drawing text, images and arbitrary geometry to the screen. It offers modern graphical features (e.g. antialiasing, 3x3 matrix, transparency, Unicode text) tightly integrated into the view and Surface systems.

Screenplay

The XML language used to describe graphics and animations. This can be used for direct graphical content, but it is also used by the View/UI system for specifying widget and application level skinning. Android widgets (e.g. buttons, lists, text-input, etc.) use this to separate their look and animated behavior from the programmatic logic. May be replaced by Adobe Flash.

SVG-Tiny Engine

Content and vector player.

Gadget Scripting Engine

A javascript-based scripting language with bindings into the system to allow "internet-style" content developers to build lite weight home-screen widgets ala Konfabulator. May be replaced by Adobe Flash.

OpenGL ES

(requires support for Java 3D as well)

Surface Flinger

A system service responsible for compositing clients' surfaces into the frame-buffer and managing the display and video hardware. It is implemented as a server that client-processes connect to. Each client is allocated one or several surfaces they can get direct access to. All the surfaces from all clients are composited by the SurfaceFlinger with optional transformations applied (blending and 2D-transform). SurfaceFlinger is designed to facilitate the use of H/W accelerated graphics through OpenGL|ES (once it becomes available) -- its feature set matches OpenGL|ES' very closely. SurfaceFlinger uses a very lightweight protocol such that no IPC is needed between client processes and itself, synchronization is done solely through atomic-operations in shared-memory segments. All frame-buffer accesses in the system go through SurfaceFlinger.

Pixel Flinger

A very low level rasterization library used by SurfaceFlinger. It implements OpenGL|ES' pixel pipeline, but none of the geometry pipeline (no 3D, no polygons, no transforms). It's a trapezoid-oriented rasterizer, which is designed to use runtime-generated code (JIT). Its API is deliberately very similar to OpenGL|ES' so that it could easily be replaced by/implemented on top of it.

Media Framework

Blah, Blah, Words, Words. See section 8 for media formats supported.

Audio

Generic "play this sound file" interface for custom sound effects. Also useful for WAV files attached to e-mails.

Streaming audio interface for longer files, such as MP3s. We may need some basic audio processing, e.g. cross-fades.

Audio mixing controls such as per-stream volume level. We need to determine the maximum number of simultaneous streams, and perhaps provide a priority mechanism if we suspect we will exceed it. (Example: playing a game with background music and 2-3 overlapping sound effects. The phone rings, or an instant message arrives. Ideally we want to reduce the volume of all of the game sounds, play the new sound, and then reset the volume. If we slow down when mixing more than 4 sounds, we need to drop the lowest-priority sound, which means knowing sound priority.)

MIDI engine for ringtones.

Microphone input. Telephony multiplexing.

Synchronized audio for stored or streaming video. This requires playing time-stamped audio and slowing or speeding audio to keep it in sync with video.

Video

Filesystem

The filesystem will be used both internally – for flash ROM or an embedded hard drive – and externally, for removable memory cards or USB-attached storage.

FAT32

Microsoft's FAT filesystem is the most common format used on removable media cards for applications like digital cameras. FAT32 supports reasonably long filenames, volumes up to 2TB, and individual files up to 4GB.

Robust file system

M-Systems driver

A Linux driver and boot loader code that supports M-Systems "DiskOnChip" product family, primarily the G4 product line which is found in many of today's smartphones. This is important for backward compatibility.

Data Manager

Asset Manager

The Asset Manager controls access to read-only assets on the device. These include graphics, sounds, ring tones, localized text, user interface definition files, and more. Assets are stored in packages based on application name, locale (e.g. “en-US”), and vendor (for vendor-specific changes). The application asks for an asset by name, and the Asset Manager takes care of figuring out which packages to search and in what order.

The packages themselves are just Zip archives, created with the Android Asset Packaging Tool. Assets may also be “loose”, stored in directories, to make it possible to update a few assets without having to send a full package over the air.

The Asset Manager does not manage downloadable content such as background images or ring tones. It treats everything as a raw byte stream.

Resource Manager

The Resource Manager sits on top of the Asset Manager, database, and filesystem. While the Asset Manager sees only streams of bytes, the Resource Manager works with typed data, such as bitmaps, localized text strings, and sound files.

SQLite

SQLite is a small C library that implements a self-contained, embeddable, zero-configuration SQL database engine.

Notification Manager

A system function that queues and alerts the user to pending system information – such as new messages, voicemail, low battery, no service, sim card alerts, etc.

Java

JVM

The VM (virtual machine) environment is the means by which bytecode-based virtual executables — Java class files in particular — can run on the system. The environment includes code to prepare and load executable files, the core class library needed by those executables, and an underlying execution engine which runs those executables. The VM environment mediates the interaction between virtual executables and the lower layers of the system by providing facilities such as a garbage-collected heap and I/O library code that wraps raw file descriptors.

The JVM is compliant to CDC/FP 1.1 (JSR 36 / 218)

Google Bytecode Interpreter

The Dalvik executable format is an executable format roughly equivalent to a Java .jar file, but with several advantages: In-memory compactness is aided by having a shared constant pool for a set of classes (rather than one constant pool per class). File parsing and verification may be done more quickly because of the use of more regularized structures, including a bytecode set which includes only fixed-width instructions. The use of a purely register-based instruction set (rather than having both registers/locals and a variable-sized stack) both simplifies verification and allows for more efficient interpretation, owing to the facts that opcode dispatch is one of the biggest sources of interpreter inefficiency and it generally requires fewer register-style instructions (vs. stack-based instructions) to perform equivalent operations. The "bytecode" code unit is 16 bits (rather than 8) because it results in fewer overall reads of the instruction stream, which is also a source of interpreter slowness. And lastly, the register-based nature of the instructions makes it much easier to have a small yet reasonably well-performing JIT.

JPDA

The Java Platform Debugger Architecture (JPDA) consists of three interfaces designed for use by debuggers in development environments for desktop systems. The Java Virtual Machine Tools Interface (JVMTI) defines the services a VM must provide for debugging (JVMTI is a replacement for the Java Virtual Machine Debug Interface (JVMDI) which has been deprecated). The Java Debug Wire Protocol (JDWP) defines the format of information and requests transferred between the process being debugged and the debugger front end, which implements the Java Debug Interface (JDI). The Java Debug Interface defines information and requests at the user code level.

MIDP 2.0

The Mobile Information Device Profile (MIDP) is a key element of the Java 2 Platform, Mobile Edition (J2ME). When combined with the Connected Limited Device Configuration (CLDC), MIDP provides a standard Java runtime environment for today's most popular mobile information devices, such as cell phones and mainstream personal digital assistants (PDAs).

MIDP 2.0 (JSR 118) is a revised version of the MIDP 1.0 specification. New features include an enhanced user interface, multimedia and game functionality, more extensive connectivity, over-the-air provisioning (OTA), and end-to-end security. MIDP 2.0 is backward-compatible with MIDP 1.0, and continues to target mobile information devices like mobile phones and PDAs.

JSR 248 – Mobile Service Architecture (Sun to provide for 2.0)

This JSR creates a mobile service architecture and platform definition for the high volume wireless handsets continuing the work started in JSR-185 and enhancing the definition with new technologies.

JSR 75 – PDA Optional packages (Part of 248 – Sun to provide for 2.0)

PIM API which gives access to the Personal Information Management (PIM) database including to-do lists, calendars and contact data. The second optional package is the FileConnection API that gives access to the mobile phone's file system including removable storage media like Memory Sticks.

JSR 82 – APIs for Bluetooth (Part of 248 – Google to provide for 1.0)

Bluetooth is an important emerging standard for wireless integration of small devices. The specification standardizes a set of Java APIs to allow Java-enabled devices to integrate into a Bluetooth environment.

JSR 135 – Mobile Media API (Part of 248 – Google to provide for 1.0)

This specifies a small-footprint multimedia API for J2ME, allowing simple, easy access and control of basic audio and multimedia resources while also addressing scalability and support of more sophisticated features.

JSR 172 – Web Services (Sun to provide for 2.0)

Blah, Blah, words, words.

JSR 177 – SATSA (Sun to provide for 2.0)

Blah, Blah, words, words.

JSR 179 – Location API (Part of 248 – Google to provide for 1.0)

A package that enables developers to write mobile location-based applications for resource-limited devices. The API works on the J2ME CLDC v1.1 and CDC configurations.

JSR 180 – SIP (Sun to provide for 2.0)

Blah, Blah, words, words.

JSR 184 – 3D Graphics (Part of 248 – Google to provide for 1.0)

Blah, Blah, words, words.

JSR 205 – Wireless Messaging API (Part of 248 -- Sun to provide for 2.0)

This defines a set of APIs which provides standard access to wireless communication resources, designed to run on J2ME configurations and to enhance J2ME profiles with unique functionality.

JSR 211 – CHAPI (Part of 248 -- Sun to provide for 2.0)

Blah, Blah, Words, Words.

JSR 226 – SVG (Part of 248 -- Sun to provide for 2.0)

Blah, Blah, Words, Words.

JSR 229 – Payment API (Part of 248 -- Sun to provide for 2.0)

Blah, Blah, Words, Words.

JSR 234 – Adv. Media API (Part of 248 -- Sun to provide for 2.0)

Blah, Blah, Words, Words.

JSR 238 – MI18N (Part of 248 -- Sun to provide for 2.0)

Blah, Blah, Words, Words

Linux Kernel

We will be using a build of the kernel from the 2.6 tree.

IPC Driver (Binder Lite)

The IPC mechanism provides standard facilities for communicating between processes. It allows a single process to host multiple distinct persistent services, which can be implemented in either C++ or Java. Communication between processes is modeled as a remote procedure call, using well-defined interfaces; this will eventually allow a tool to automatically generate the necessary marshalling code for all languages based on an abstract interface definition. A custom kernel module (derived from OpenBinder) is used to implement inter-process calls, with features such as object mapping between processes, directly dispatching calls to a pool of available threads in a process, and performing a single copy of the data going between processes. The high-level APIs included for communicating across process, in both C++ and Java, look similar to COM's interface-based design.

Keypad Driver

Linux driver to control the typical phone keypad, including numeric (0-9, *, #) D-Pad and call control functions.

Display Driver

Linux driver to control the typical LCD display, both primary and secondary (may be separate drivers). Includes backlight and low power modes.

Baseband Driver

Linux driver to interface to various semiconductor companies baseband core.

802.11 Driver

Linux driver to interface to various semiconductor companies wifi baseband core.

SD/SDIO

Flash Driver

Linux driver to interface to various semiconductor companies NAND flash chips. See section 4.4.3 for M-System flash driver.

USB Driver

GPS Driver

Linux driver to support standard interfaces for various GPS and A-GPS chips. Optional per minimum hardware requirements.

Power Management

System will support low power mode and processor clock cycling. Drivers will support subsystem shutdown where appropriate, and cleanly be able to power up individual subsystems quickly. System will strive to support industry standard talk/standby times.

Camera Driver

Linux driver for support of CMOS modules by vendors such as Omnivision and Micron.

LIBC

4 Communications

Blah, Blah, Words, Words.

Telephony APIs (TAPI)

The telephony subsystem provides a high-level API to the traditional mobile phone features on the device. It will be portable across various GSM baseband software implementations, communicating with them primarily with the 3GPP TS 27.007/27.005 AT command sets over either a TS 27.010 (CMUX) style link or a manufacturer-specific link where appropriate. The middle layer is designed to be portable to CDMA-style commands as well, although CDMA is not an initial target. High-level API's to SMS are made available, and packet-switched data will be routed to the Linux TCP/IP stack and made available through traditional Berkeley sockets APIs.

Baseband Downloader

Code that runs at boot-loader time that copies the baseband firmware to the proper memory address and resets that baseband controller to start executing the code. Also responsible for configuration and parameter adjustment for various RF front end devices.

TCP/IP

Provided by Linux

PPP

Provided by Linux

Data session manager

```
The Connection Manager is a policy agent for managing packet-switched
data connections. The Connection Manager's goal is to provide an
```

experience as near to "always on data" even in situations where carrier requirements or billing schemes do not make true always-on data reasonable. This near-transparency will extend as much as possible both to the user interface and to the application. The Connection Manager is intended to allow the user/subscriber to view and budget their data usage without discouraging their use of data services. Of course, the Connection Manager will allow an "always on" data mode for accounts where that class of service is appropriate. Will also support multiple paths and arbitrate between GSM and WIFI.

Bluetooth

Bluetooth is a wireless communication protocol mainly used for short distance and in devices with low power consumption. Because Bluetooth is capable of communicating in an omni-directional manner of up to 30 feet at 1 Mb/s it is far superior to infrared.

Host Controller Interface (HCI)

The Host Controller Interface (HCI) provides a standard interface to the Bluetooth baseband controller and link manager services that is independent from the host hardware implementation. This layer provides a uniform method of accessing any Bluetooth hardware. There is an addendum to the HCI specification for different host transport protocols. For each physical bus (USB, RS232, UART etc.) it defines the interface functions based on which physical bus is used, but also vendor specific implementations are possible. Support for H2 (USB) and H4(UART) as required by chipset.

Logical Link Control and Adaptation Protocol (L2CAP)

The Logical Link Control and Adaptation Layer Protocol (L2CAP) is layered over the HCI layer. L2CAP provides connection-oriented and connectionless data services to upper layer protocols. It uses a Protocol and Service Multiplexing (PSM) capability and a Segmentation and Reassembly (SAR) mechanism. L2CAP permits higher level protocols and applications to transmit and receive L2CAP data packets up to 64 kilobytes in length. Also group abstractions and Quality of Service (QoS) features are supported.

Serial Port Emulation (RFCOMM)

The RFCOMM Protocol provides emulation of serial ports (RS232) over the L2CAP layer. The protocol is based on the ETSI standard TS 07.10. Only a subset of this standard is used and some adaptations are Bluetooth specific and are documented in the RFCOMM specification.

The serial emulation enables compatibility with a large base of applications that currently use the serial port as their main communication bus. RFCOMM conveys all of the RS232 control signals and supports remote port configuration. While the protocol is based on an ETSI specification, the name is related to IrComm of the IrDA protocol stack.

Service Discovery Protocol (SDP)

The Service Discovery Protocol (SDP) provides the means for client applications to discover the existence of services provided by server applications as well as the attributes of those services. The attributes of a service include the type or class of service offered and the mechanism or protocol information needed to utilize the service.

SDP involves communication between a SDP server and a SDP client. The server maintains a list of service records that describe the characteristics of services associated with the server. Each service record contains information about a single service. A client may retrieve information from a service record maintained by the SDP server by issuing a SDP request. If the client, or an application associated with the client, decides to use a service, it must open a separate connection to the service provider in order to utilize the service. SDP provides a mechanism for

discovering services and their attributes, but it does not provide a mechanism for utilizing those services.

Normally, a SDP client searches for services based on some desired characteristics of the services. However, there are times when it is desirable to discover which types of services are described by an SDP server's service records without any a priori information about the services. This process of looking for any offered services is called browsing.

K2 – Service Discovery Application Profile

This profile defines the features and procedures for an application in a Bluetooth device to discover services registered in other Bluetooth devices and retrieve any desired available information pertinent to these services.

Essentially, the service discovery profile defines the protocols and procedures that shall be used by a service discovery application on a device to locate services in other Bluetooth-enabled devices using the Bluetooth Service Discovery Protocol (SDP).

K6 – Headset profile

The Headset profile defines the requirements for Bluetooth devices necessary to support the Headset use case. Essentially the Headset profile defines the protocols and procedures that shall be used by devices implementing the usage model called 'Ultimate Headset'.

K7 – DUN profile

The Dial-up Networking profile defines the requirements for Bluetooth devices necessary to support the Dial-up networking use case. Essentially the Headset profile defines the protocols and procedures that shall be used by devices implementing the usage model called 'Internet Bridge'. The most common examples of such devices are modems and cellular phones.

Two main scenarios are implemented: the Usage of a cellular phone or modem by a computer as a wireless modem for connecting to a dial-up internet access server, or using other dial-up services , and Usage of a cellular phone or modem by a computer to receive data calls

5 Phone Functionality

Dialer

Call History

Feature exists to simplify adding new contact info and dialing. Incoming and outgoing calls must be tracked and recorded. It will be used by other features such as phonetop search (see section 11.4)

Contacts

Synchronized with Google services and support G-Talk presence (see section 11.5)

Calendar

Synchronized with Google services (see section xx.x)

World Clock

With Alarm.

Settings

Ringtone Management

Should include a mechanism to switch between Silent, Meeting, Vibrate, etc.

Telephony setup

Data setup

...

Skinning/Wallpaper

Camera

Still

Video

Media Player

Audio

Video

Download Manager

SMS

OMA MMS 1.2 / 3GPP TS 23.140

Multimedia Messaging Service (MMS) is a system application by which a client is able to provide a messaging operation with a variety of media types: WBMP, GIF87, GIF89a, JPEG, PNG, BMP, AMR, MIDI, WAV, H.263, SVG, MPEG4

SMIL

Synchronized Multimedia Integration Language. SMIL is a presentation format, i.e. a SMIL page contains information about the appearance of different multimedia elements on a display. When SMIL is used to represent content on a PC screen, normally a window is opened whose size is defined by the layout element of the SMIL page to be displayed. In this way, the appearance of the SMIL page on the screen will reflect exactly the organization of the content as the author had created it. When SMIL is used for the presentation of multimedia messages on mobile terminals, the size of the window is severely limited by the resolution and appearance of the terminal display. The layout of a multimedia message represents the content as created by the originator, but it is well possible that the original layout simply does not fit into the display of the receiving terminal. Therefore, SMIL exchange must be simple enough to ensure that -if the displays of the originator and receiver terminal are different- the content can still be displayed, possibly by changing the relative position of the different elements.

WAP 2.0

With backward support for 1.x

Markup

WML (238)

WCSS 1.1

WML Script (193)

XHTML (277)

WBXML

Adjunct Layers

Push (235)

WTLS – security

Media Types (237)

WBMP, GIF87, GIF89a, JPEG, PNG, BMP

Protocol Stack (jWAP sourceforge)

HTTP

WSP (203)

WTP

WDP

SIM toolkit

Blah, Blah, Words, Words.

6 Synchronization

Blah, Blah, Words, Words.

OBEX

OBEX is a high-level API and protocol for exchanging objects such as electronic business cards and calendar items transmitted in the vCard and vCalendar formats. OBEX was originally introduced by the Infrared Data Association (IrDA), and it can be implemented on top of the Bluetooth RFCOMM protocol, IrDA protocol, TCP/IP, and others.

Data Management

OMA DL

The OMA Download (OMA DL) specification provides a flexible protocol for enabling the download of generic content, controlled through the use of a separate download descriptor. By drawing on elements of these protocols and adding new elements, an effective protocol is constructed that combines Device Management for controlling the main device configuration management functions and provides for the use of descriptor-based download mechanisms to download larger binary objects such as firmware updates. The download process is abstracted to allow the use of either OMA DM (e.g. Add/Replace - In-Session method) or any suitable alternative download mechanisms (for example, a descriptor based download protocol such as OMA OTA Download v2.0 - Alternative Download Method).

OMA DM

Device Management refers to the process of managing device settings and applications resident in the terminal. Device Management facilitates the configuration of the user's terminal with minimal user intervention to enable easy usage of mobile services. As terminals offer more and more services, Device Management provides a mechanism for the operator to remotely configure a terminal's parameters while the terminal is in the hands of a user.

OMA DS

OMA Data Synchronization (OMA-DS) is an open standard that defines a common protocol used to synchronize networked data between two or more devices to keep the data current and consistent. The OMA-DS client stack provides data synchronization of personal information such as calendar, contacts, tasks, and email. Synchronization ensures that any changes made to the calendar in one device, for example, are also reflected in the calendar in the other device.

SyncML

An industry standard synchronization protocol and data format that works in conjunction with OMA DS and OBEX. Developer will consider supporting MSFT ActiveSync support as well.

7 Media Formats

Image Formats

PNG

PNG (Portable Network Graphics) is a lossless compressed image format. It uses the zlib library, and is free of patent restrictions. It is widely used on the Internet. We will use the open-source "libpng" library to manipulate them.

JPEG

JPEG (Joint Photographic Experts Group) is a lossy compressed image format that is mostly free of patent restrictions. It is the most widely used image format for digital pictures. We will use the open-source IJG JPEG library to manipulate them.

GIF

GIF (Graphics Interchange Format) is a lossless compressed image format suitable for 8-bit images. It uses the LZW compression algorithm, for which the patents have expired. It is very popular on the Internet for small images on web pages and the wireless industry has adopted it for various standards, such as WAP and MMS.

BMP

BMP (BitMaP) is the lossless bitmap format developed for OS/2 and made popular by Windows. It's not used often on the Internet, but it's a common format among developers and the wireless industry has adopted it for various standards, such as WAP and MMS.

Audio CODECs

- 3GPP-Rel6
- AAC, AAC+
- AMR-NB, AMR-WB
- MP3
- PCM a–law, u-law

Video CODECs

- 3GPP-Rel6
- H.263
- H.264
- MPEG4

8 Tools

APROTO – Remote debugging tool

IDE support

- Eclipse
- NetBeans

UI authoring tool (XML editor)

Simulator

The simulator front-end provides a user interface for developing and testing handset software. Handset specifications, which include pictures of the device, lists of available buttons, and the resolutions of displays, are loaded from config files. The front-end talks to the device code, which can be compiled to run natively on the desktop system or (someday soon) built for ARM and emulated. Keyboard input and physical state changes, such as "open flip phone", are passed into the simulated device, and display updates are sent out. The front-end was developed with the wxWidgets UI toolkit, and runs on Linux, Mac OS X, and Win32.

ARM Emulator

The ARM emulator ("armsim") is a software tool for simulating executables that are compiled for the ARM instruction set. Armsim can simulate both stand-alone user programs as well as the Linux kernel. When simulating the Linux kernel it also models the MMU (for both instructions and data), the interrupt hardware, and devices such as the UART (for keyboard input) and the LCD display device (not completed yet). When simulating standalone user programs, armsim sets up the stack with command line args and environment variables before simulating the user code. Armsim can also produce a detailed instruction trace. The trace contains every single instruction executed (including interrupt handlers in the Linux kernel) and is useful for debugging performance problems and for getting very detailed information about what is happening inside a program.

Testing

We need to provide a set of regression tests for various features. [How will this work? Does this deserve its own full section? Should it fold into SDK, or be a sub-section of the earlier sections?]

[section on: Java compliance testing]

[section on: other industry compliance stuff? Maybe that just goes with the relevant standards in the telecom section.]

9 HTML Web Browser

Android Browser is an implementation of Apple's Webkit open source. Apple's Webkit consists of WebCore (an html rendering engine), JavaScriptCore (the Javascript interpreter), and WebKit (a browser embedding API). We will port WebCore and JavaScriptCore to the Android platform. We won't use WebKit as is, and will build our own flavor of it, and probably provide a Java browser embedding API.

We'll need to build mobile specific features on top of the WebKit solution, such as Small Screen Rendering (SSR) for small screen layout, constrained device memory management, mobile browser link and page navigation, connection management suitable for low bandwidth/high latency network. We will also need to add cache and cookie management for constrained memory environment.

In addition, we hope to innovate in the area of ease of use. We'd like to improve the url entry on phones, with shortcuts, type ahead lookups, and potentially better integration with Google search.

There are other features that we'd like to have, but may not be able to get to for 1.0 ship. Nokia has done some interesting UI on thumbnail navigation, which makes 2D navigation easier on pages laid out wider than the phone screen. We think it will be an interesting alternative to SSR layout.

Lastly, we need to provide the html rendering engine as a library, so other apps besides the web browser that want to render html content can use it.

Web Browser Acceleration Service

Blah, Blah, Words, Words.

Proxy Image Transcoder (under consideration)

Download of images in their original sizes can be wasteful. If the image width is bigger than the screen width, the browser will scale down the images in order to fit them in the layout. So the user may never need to see the images in their original sizes. With a proxy server, it can scale the images down before sending them over the wireless network. In addition, the proxy transcoder can strip out unnecessary information in image formats such as JPEG and GIF (i.e. text comments), and reduce the color depth if the mobile phone doesn't support all the colors in the image.

This is an optional feature should the wireless carrier choose not to enable it.

Connection Proxy (under consideration)

Wireless networks have high latency and low throughput. Neither is desirable for download speed. Having a proxy server can help resolve some of these issues, as the proxy server can keep a persistent socket connection open and multiplex several streams to avoid the high cost of

setting up new socket connections over the wireless link (3-way TCP handshake: SYN, SYN-ACK, DATA + ACK bits) and teardown (FIN, FIN-ACK + RST)

This is an optional feature should the wireless carrier choose not to enable it.

10 Google Specific Features

Some of these are *optional features* should the wireless carrier choose not to enable them. This, however, does not imply that a subset of these applications will be developed. The entire list will be developed and released with the system. The features in this category must be deeply integrated with the system and have a seamless experience.

Google Download

A high-fidelity content distribution system. Will support industry standards such as OMA DL, and will focus on user experience and ease of transaction. Features such as "try before you buy" for both applications and media are important to consider. They will help differentiate the Google download system from competitors.

G-Mail

GLM

Phonetop Search

G-Talk

Instant Messaging only. Jabber XMPP support.

G-Cal

Synchronized with a central server.

Contacts

Synchronized with a central server.

Sputnik

Embeddable Sputnik Gadget Scripts (see section 4.2.17) that speak the protocol to the Sputnik server.

Lighthouse Photo Upload

Only support for photo uploading.

11 Intentionally left out

The following features and functionality have been purposely left out of the 1.0 release. The reason for dismissal is recorded under each section.

Google Advertising Engine

Reserved for a future release

Push To Talk

Perfect opportunity for a third party developer

IrDA support

Either IrDA or Bluetooth, but why both?

Presence

Section refers to IMS-style presence. Contacts will include presence for G-Talk (see section 11.5) Perfect opportunity for a Commercial implementation

Voice Dialing

Text To Speech

iGoogle

Perfect opportunity for a Google-team

Orkut

Blogger

Lighthouse

Will include support for Neven Vision code for facial feature tracking and detection. System will make these features available to third party developers to automatically embed image metadata within the exif header so Google services can take advantage of information taken at point-of-capture (such as location, pose, number of faces, etc.)

Enterprise sync

Initial release is targeted toward consumers. Perfect opportunity for a third party developer

Behind the firewall e-mail

Initial release is targeted toward consumers. Perfect opportunity for a third party developer

Various JSRs

Such as JSR 248. Perfect opportunity for a Commercial implementation. (See section 4.6.5)

Proprietary Instant Messaging standards

AIM, MSN, Yahoo

Games

Perfect opportunity for a third party developer

EMAIL

Perfect opportunity for a Commercial implementation and may be required by carriers. POP and IMAP email reader. Must include support for SSL. Must also support a reasonable set of MIME converters for popular enclosure formats, such as Word and PDF.

On Apr 20, 2006, at 1:01 PM, Rich Miner wrote:

> do you have a more recent PRD spin?

EXHIBIT J

From:	Dan Bornstein.	Sent:6/12/2006 1:29 PM.
To: [-]	Android Engineering [. .].	
Cc: [-]	.	
Bcc: [-]	.	
Subject:	Re: [Android-eng] String::split.	

Yes, enums are relatively expensive compared to static final ints.
That being said, I think the right place to draw the line is at the
API boundaries. That is, if it's for something to be presented to
programmers, err on the side of using an enum. If it's just internal
plumbing, and efficiency is paramount, then use static constants.

In terms of all the new Java 5 language features, I'm pretty sure the
only one that requires cooperation from the VM is the runtime
annotation facility. (And we probably won't have runtime annotations
any time soon.) Everything else is syntactic sugar, with a bit of
library support (e.g. and already mentioned, Iterable). If you want to
use a Java 5 language feature, then I have no problem with you doing
so, but be prepared to implement missing pieces out of our java.*
hierarchy. And if you run into trouble that you don't feel qualified
(or compelled) to fix, then please submit a bug.

And, as previously mentioned, the new features generally do come with
a runtime cost, so keep that in mind. For example, you probably don't
want to use a collection iterator ("for (Blah var : coll) ...") in an
efficiency-critical inner loop. I'm personally a big fan of using
profiling to figure out when I can no longer get away with writing
straightforward code.

As for what set of java.* classes we're aiming for, that's still
somewhat of an open question, the resolution of which will undoubtedly
hinge on what happens (or fails to happen) with Sun. That being said,
I think CLDC compliance is a reasonable baseline, and if we could hit
the CDC foundation, that would be awesome. But if there's some java.*
functionality you need that isn't in either of those, it's still okay
to go ahead and implement it (or file a bug for same). If we have to,
we will play classloading tricks to make sure that the restricted
environments (such as CLDC MIDLets) only see the classes that they're
allowed to.

-dan, your Java guy

EXHIBIT K

From:	Dan Morrill.	Sent:6/25/2007 12:18 PM.
To: [-]	Dan Bornstein.	
Cc: [-]	.	
Bcc: [-]	.	
Subject:	Re: Android JRE Coverage.	

Very interesting info. It sounds like I'll just have to suck it up for a while, yet.

Ultimately, the most important question is: will JRE-completeness (according to some metric, at least) be gating for public launch? I can only assume "yes". :)

- Dan

On 6/25/07, Dan Bornstein <danfuzz@google.com> wrote:

On 6/25/07, Dan Morrill <morrildl@google.com> wrote:
> I have another question for you -- Mike Cleron said you could help me out
> with this, too. (If there's a list I ought to sign up for, let me know.)

I'm a fine person to ask these questions of, but if you want to ask
them in a more open forum, there is the list
<android-googledevelopers@google.com >, which is -- as the name says --
for Google folks in general who are doing development on the Android
platform.

> Since I had assumed that Android was implementing J2SE, I am obviously wrong

Actually, you aren't *that* wrong; you are merely working with an
incomplete system.

> What is the JRE-coverage strategy for Android? Is the intent to support
> some Java standard configuration? If so, which? (J2SE? J2ME/MIDP? etc.)

Our nominal goal is to achieve approximate parity with CDC Personal
Basis Profile, which has most of the classes from SE that one could
reasonably expect to be useful on a high-end mobile device of recent
(or soon-to-be) vintage.

Unfortunately, though, due to our situation with Sun, neither I nor
any of my Android compatriots are allowed to refer to any documents
from Sun that require actively accepting a license (including
click-through), so we can't actually attempt to exactly hit CDC PBP.
It's more like our goals and the reasoning behind the PBP spec are
similar, so the results of applying those reasons will also be
similar.

> What is the timeline for attaining this coverage?

Much of the library implementation work is being done by an outside
vendor, and they have a schedule which currently states completion by
the end of September. For various reasons, I don't believe that date.
However, there are intermediate milestones, and with regards to your
specific stated needs, java.util is coming soon (I have an alpha drop
from them which is sitting in a branch), but I don't expect to see any
XML-related stuff for at least a month.

> Are there any reasons not to suck in Apache Harmony classes, to fill in the
> gaps?

Actually, much of what the vendor is doing is sucking in classes from

Harmony, and adapting them to work (well) on Android. But even this takes time!

> will the J2ME classes be supported too?

Sorta. J2ME compatibility won't be built into the main system but will be (optionally) available as a separate add-on. There are a few reasons for doing it this way, but the upshot is that you won't be able to write an Android app while also using J2ME-specific classes. (Sorry.)

-dan