Slide 1:
My name is Terrence Barr, I am a Senior Technologist for Oracle.

I am here today to talk about Oracle's strategy for mobile and embedded Java. At JavaOne 2011 Oracle announced its strategy, roadmap, and products for mobile and embedded Java. In today's presentation I would like to review these announcements with you and give you a little more detail and outlook.

Slide 2:
Please observe the Safe Harbor Statement.

Slide 3:
Java is big. It covers a vast range from massive servers all the way down to tiny smart cards. What is common among all these Java platforms is the Java language. Depending on the specific environment we distinguish between the Java SE platform, the Java ME platform, and the Java Card platform. For each specific area there are also key APIs on top of the underlying platform. In the server space we have Java EE, on the desktop Java FX, and the embedded space has different verticals with their own specific APIs like Java TV, BD-J, and others. The mobile space has it's own set of APIs, such as the Mobile Services Architecture or MSA.

Slide 4:
The key point here is that Java technology runs across the entire spectrum of platforms. This is a powerful message because it makes software development much easier and significantly improves time-to-market. Other technology platforms in the mobile and embedded space are much more fragmented and you cannot easily transfer your skills or code between them, often requiring steep learning curves. As you will see throughout this presentation Oracle is focusing on bringing the benefits of Java everywhere in a homogeneous and scalable manner.

Slide 5:
To understand the direction that Oracle is taking let's have a quick look at the core Java platform, which is Java SE. Java SE has six major design objectives. "Run Anywhere" is something we just talked about. Oracle plans to add new application models to Java, as well as support new hardware in order to make Java available for particular use cases. Another major design objective is to make the Java runtime smaller and more flexible through modularization. Next, is to constantly improve productivity for Java developers. The Java platform is also increasingly the becoming the premier choice for languages other than Java, such as scripting languages. Finally, Oracle is constantly seeking to improve runtime performance, monitoring, and diagnostics.

These design objectives underly not just Java SE, but the Java platform as a whole. As you will see this is also reflected in the strategy for mobile and embedded Java.

Slide 6:
Before we go there I would like to briefly talk about the achievements in the Java products throughout 2011, in particular regarding Java SE, JavaFX, and Java ME. Oracle delivered JDK 7 as promised, which really got the whole Java platform moving

forward again. JavaFX 2.0 was also released, and as you will see JavaFX will be a key technology not only on the desktop but also in the embedded space. Finally, a number of products and updates were released for Java ME. Oracle Java Wireless Client (OJWC), Oracle Java Embedded Client, Java SE for Embedded, the Java ME SDK, and the Lightweight UI Toolkit (LWUIT) were all released in 2011. Oracle has really started up execution again and is making progress across the entire Java product line.

Slide 7:
The Java Community Process or JCP is the standardization mechanism for Java technology. The JCP is moving forward again and a number of new JSRs, or Java Specification Requests, have been filed or will be filed soon. Also, the elections for the JCP Executive Committee (EC) just completed. ARM Ltd. is of course a very important entity in the mobile and embedded industries and we are happy to welcome ARM as a newly elected Java ME Executive Committee member. This also underscores the growing importance of mobile and embedded Java to key players in the industry.

Slide 8:
Let's double-click on the Java spectrum for mobile, embedded and card. First, a bit of history: When Java 1.0 was released it was a single platform with a single set of libraries, a single runtime, and a language with a specific set of features. Over time the Java runtime was tailored to address the needs of specific markets and devices. So the Java platform diverged into different stacks, each of which had specific features. Java Card has a very restricted feature set, the CLDC stack, which is mostly used in mobile devices, has a slightly larger feature set. The CDC stack, which is used in many consumer devices, is yet another stack. And all of these are more-or-less strict subsets of Java SE. While this made sense at the time it is now causing problems for developers because code and skills cannot easily scale across the spectrum of Java platforms. What Oracle wants to achieve is to make Java platforms much more uniform in terms of the runtime, the libraries, and the language features for the benefit of developers and the Java ecosystem.

Slide 9:
So now let's talk about the design objectives for Java ME moving forward. The 1st objective is to bridge the divide I just spoke about between Java SE and Java ME. Oracle intends to achieve this in 2 ways. First, by synchronizing the CLDC and JDK releases. Second, by converging the CDC stack and Java SE Embedded.
The 2nd objective is to have full coverage of embedded vertical markets. This means that the Java platform can cover all relevant CPU and footprint variants. Also it is planned to add dedicated APIs for important vertical market segments.
The 3rd objective is to add deep integration of content and services to the Java products. This means embedding value added services into Oracle runtimes and tools to enable developers and consumers easier access to carrier features.

Slide 10:
Let's double-click on this. Bridging the Java SE/ME divide is really about 2 aspects. Creating the future of Java ME with Java ME 7 and 8 and the CDC/Java SE 8 convergence.

Java ME 7 and 8 encompasses two aspects: 1st, aligning Java ME with Java SE. This means synchronizing the releases of the respective platforms, adding the latest Java language support to Java ME, implementing selected Java ME APIs on Java SE, and finally ensuring consistent tool interfaces between Java ME and SE. 2nd, Oracle plans to add new APIs for mobile phones and billions of connected devices to make the creation of content and the consumption of services easier and more compelling.

In terms of CDC/Java SE 8 convergence, were talking about 3 aspects. Instead of being it's own stack and product, the CDC platform will become a profile in Java SE 8. This includes porting features of the CDC virtual machine to the Hotspot JVM as well as creating JDK 8 libraries which are optimized for embedded with respect to size and speed. Also Oracle plans to use JavaFX as the graphics framework for mid-to high-end embedded platforms. As a result of these efforts developers will get the best of both worlds in one package, meaning the Java SE platform will scale seamlessly across a wide range of devices.

Slide 11:
With that let's have a look the segments for embedded Java within the next two years. In the low end segment, up to about 1 MB, you will find Java Card technology, which has really been designed for that space. For small embedded, which covers a footprint of roughly 1 MB to 10 MB, Java ME will be the platform of choice. The remainder of the embedded space, from midrange to high-end from roughly 10 MB upward, will be covered by Java SE. For processors, Oracle plans to support a full range from the low-end ARM 7, continuing with cortex M and ARM9/11, all the way up to MIPS32, Atom, PPC, and CortexA.

In summary you can see that Java will scale over the entire breadth of the embedded market in terms of language support, runtimes, operating systems, and processors.

Slide 12:
in 2011 we have seen a number of developments for Java in the mobile and embedded space. In Java card, near field communication, payment, and E passport, as well as machine to machine (M2M), and embedded security. More than 3 billion phones and 125 million TVs are currently running Java and new features such as web integration and new UI technologies are coming to these markets. In embedded, we are seeing more and more embedded applications connecting back to the enterprise, for example media and Web server streaming and new applications in the entertainment space which leverage back-end content and functionality.

Slide 13:
As I mentioned before, the integration and consumption of new services on mobile devices is critical to creating compelling content that developers can monetize. To that end, Oracle is adding new features to its mobile Java offering called Oracle Java Wireless Client (OJWC). OJWC 3.1 will have support built-in to simplify payment, access backend content stores, integrate advertising, as well as APIs to access carrier functionality. Mass-market mobile devices present a huge distribution and monetization opportunity for developers, and OJWC will make it easy for developers to create and sell to these markets.

Slide 14:
Oracle is making good on its commitments and in 2011 delivered a host of new or updated products such as the Oracle Java Wireless Client (OJWC) 3.1, the Oracle Java Embedded Client 1.x, LWUIT 1.5, Java ME SDK 3.0.5, Java SE Embedded 7, and a number of others.

Slide 15:
As an example of one of the features which Oracle delivers with OJWC 3.1 I would like to discuss the mobile network APIs. These are a set of pre-integrated APIs that enable operators to expose capabilities from the networks to developers. These are features such as messaging, call control, user profile, payment, location, authentication, and more. To make this more useful, the server-side application can access some of the mobile capabilities as well. The APIs follow industry standards and is architecture neutral. On the next slide, let's have a look at how this works.

Slide 16:
The Java ME platform provides some basic building blocks and on top of these are APIs for each specific piece of functionality. The binding to the particular implementation for a network operator is accessed through these interfaces. In the code section you can see a class which implements the OperatorAPI for operator ABC. The benefit of this approach is that developers can rely on high-level functionality built into the platform to leverage easily from any application.

Slide 17:
in the embedded space, such integration with external services is already a reality, for example on the Blu-ray platform. Every Blu-ray player ships with a Java runtime, and Java applications can access external services, for example DLNA services or back-end servers to create more interesting interactive experiences. In this example, the Blu-ray movie "Rio" contains an interactive Java application called “Postcards from Rio” which uses media assets from the movie and interacts with external DLNA-enabled devices such as printers.

Slide 18:
So what are the high-level milestones for Java ME and embedded Java over the next 2 years? 2011 saw the release of OJWC 3.1. In 2012 Oracle and its partners will be busy with Java ME 7, and in 2013 we are targeting delivery on the CDC/Java SE convergence. After that, we are planning for Java ME 8 and beyond.

In the meantime, of course, we are not sitting still and will deliver incremental updates to CDC and Java SE Embedded. Also, the latest versions of the developer tools are available, so you can download Java ME SDK 3.0.5 and LWUIT 1.5 and get started today.

Slide 19:
So let's wrap up. The Java platform is moving forward and we're bringing Java together again by aligning Java ME with Java 7. We're adding new productivity features, new APIs, modularity, profiles, and more platforms to Java, allowing developers to code

anywhere and run on tiny embedded to massive servers. We're updating products, solutions, and content with OJWC 3.1, Java Card 3.0, LWUIT 1.5, and more. And finally, be sure to check out the latest ready-to-run releases of embedded Java, including Java SE 7 for Embedded on ARM and X86, as well as Oracle Java Embedded Client (OJEC).

Slide 20:
Java is moving forward and picking up steam ... Stay tuned.