
JavaOne™

MAKE THE FUTURE JAVA

ORACLE

ORACLE

Expanding the Reach of the Java ME Platform

Roger Riggs
Consulting Member of Technical Staff
Andrey Petushkov
Member of Technical Staff

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Program Agenda

- Java ME 8 Platform
- CLDC Alignment with Java SE
- Java ME Embedded Profile
- Device Access API
- Java ME 8 Support for Mobile
- Looking Ahead to Java ME 9

ME Evolution : Towards a Common Java

Key Principles

- ME is the “little sibling” of SE.
- CLDC is a strict subset of SE
- ME applications and libraries work across the Java Platform
- ME vs. SE is a footprint/functionality tradeoff
- ME & SE release cycles are in sync

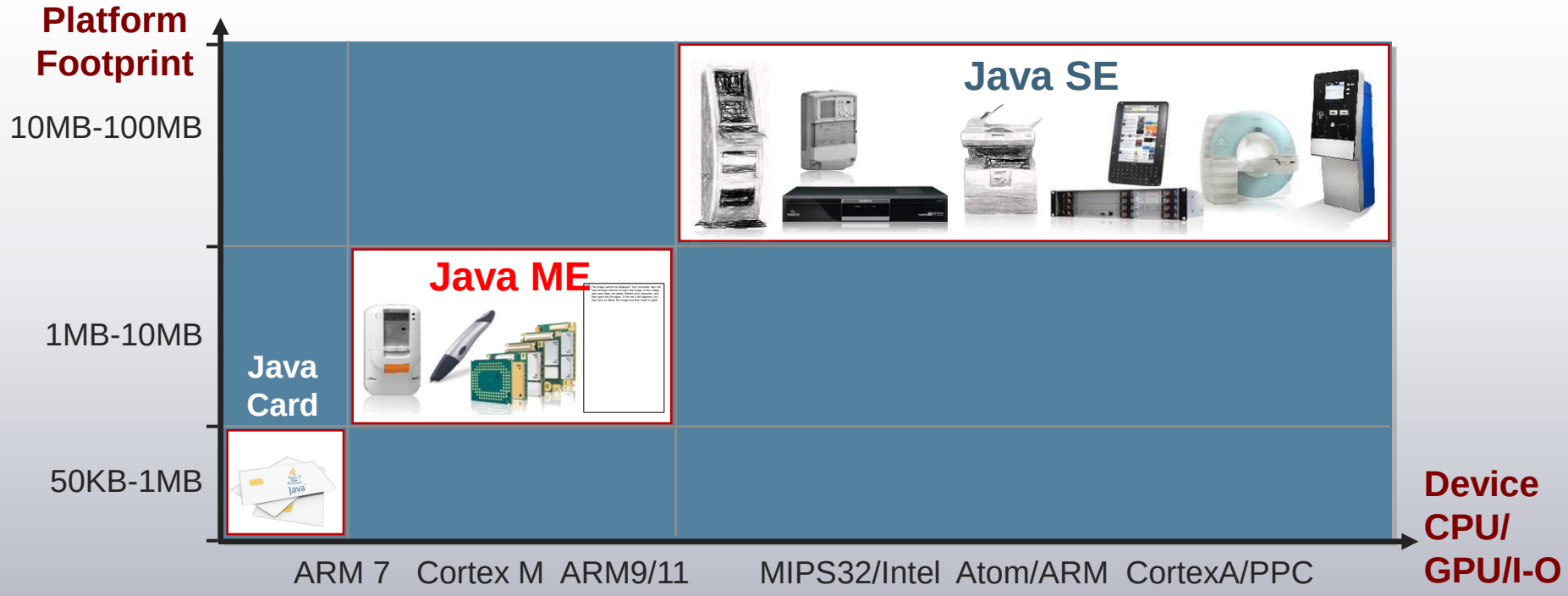
Benefits

Unified development experience & community
Align language, core APIs, development and management tools
Enable value in SE by reusing ME APIs : Bluetooth, Location, Sensors, Messaging ….
Enable value in ME by reusing SE tools and management features
9+ Million developers for all of Java, from Card to Enterprise, to have access to Java ME ecosystem

Market Driven Strategy

- Java ME 8 increases the reach of Java ME to a broader range of small connected devices - beyond feature phones
- Mobile developers will benefit from an expanded accessible market and access to a broader community and expertise
- Mobile OEMs can leverage the new feature in ME 8
  - Alignment with JDK 8 language and tools, services framework, shared libraries, events and notifications, AMS APIs etc...
- Java ME 8 will offer building blocks options for OEMs to derive compact and performant runtimes based on ME 8

Java Technology for Embedded Device



Enabling the next billion

The Java ME 8 Platform Release

The first step towards a common Java

- **Release Themes**
  - Platform modernization
  - ME / SE Language Alignment
  - Standardized Embedded API
- **Key Features**
  - CLDC 8 to align language/tooling with SE8
  - Dedicated APIs for small embedded - Java ME Embedded Profile and the Device Access API JSRs
  - Improved on-device tooling
- **Target Markets**
  - Small Embedded (M2M edge devices, communication modules, healthcare monitoring devices, smartmeters)

What is Java ME 8?

A modernized cross device Java ME platform

- ME 8 represents a critical modernization of the Java ME Platform
  - VM / Profile functionality
- ME 8 will target a broad range of small embedded devices
  - M2M modules, microcontrollers, medical monitoring devices, etc.
  - Feature phones are one supported vertical market
- ME 8 platform should be usable across market segments
  - Modular / optional specs instead of vertical feature sets

Java ME Platform Architecture




Vertical Specific APIs

Other Vertical Specific APIs…	Wireless Messaging	SATSA	Vector Graphics	Content Handler
	Bluetooth	Media	3D Graphics	Sensors
Location	Contactless Communication	XML	OpenGL 1.1	Personal Info

On-Device I/O Peripheral Access: Device Access API (aka DA API)

Application Environment: Java ME Embedded Profile (aka IMP 8)

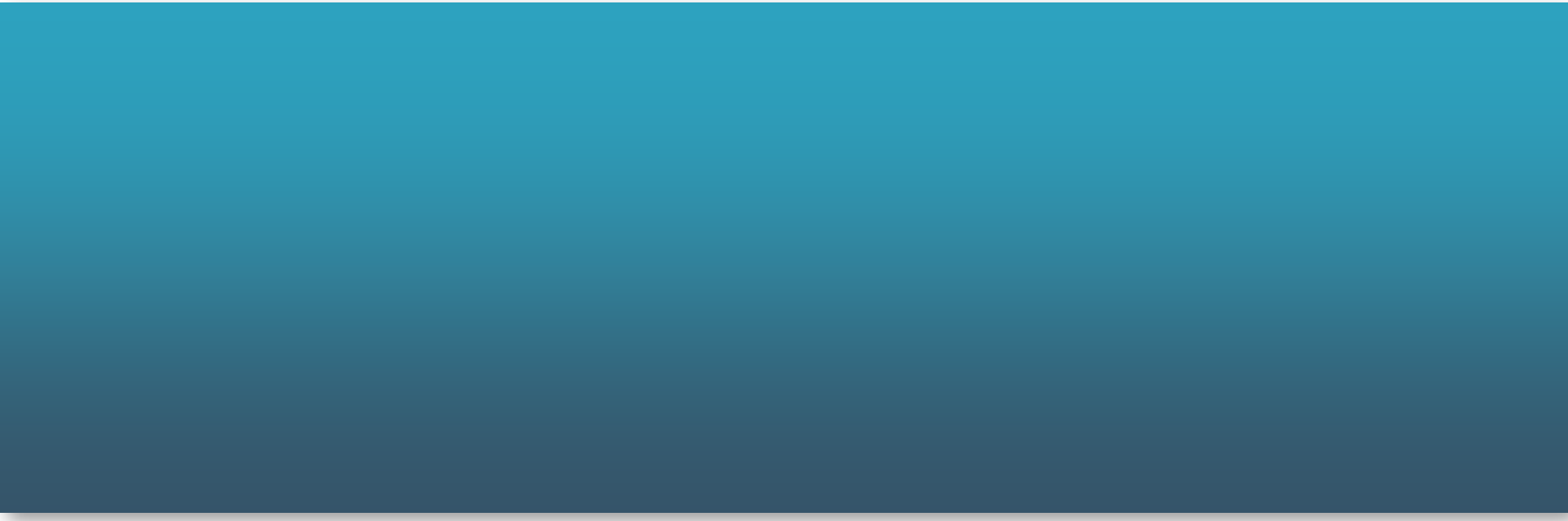
Java VM: CLDC 8 (with Support for Java Language 8)

Two New JSRs Submitted

- Connected Limited Device Configuration 8 JSR Proposal – JSR 360
  - Initial EG members Aplix, Nokia, North Sixty-One, Stefano Andreani
  - Supporters: ARM, Deutsche Telekom, Siemens, TOTVS, Vodafone
- Java ME Embedded Profile – JSR 361
  - Initial EG members: Aplix Corp., Cinterion, Deutsche Telekom, North Sixty-One Ltd, Stefano Andreani
  - Supporters: ARM, Siemens, TOTVS, Vodafone
- Please Review the Proposals and comment
- Plan how you can contribute!
- http://jcp.org

Program Agenda

- Java ME 8 Platform
- CLDC 8 Aligning with Java SE
- Java ME Embedded
- Device Access API
- Java ME 8 Support for Mobile
- Looking Ahead to Java ME 9



CLDC 8 Overview

- Java Language and libraries in alignment with Java SE 8
- Key Features
  - Synchronize with Java SE 5/6/7/8 Language Features into ME
  - Virtual Machine Update
  - Remain as small as possible - footprint optimizations
- Specification Requirements
  - CLDC to be an extended strict subset of SE
  - Backward binary compatibility
  - Consolidated Generic Connection Framework

Java Language Features (New to CLDC)

- Assertions
- Generics
- Enhanced for Loop
- Autoboxing
- Enumerations
- Varargs
- Static imports
- Annotations

- Java SE Features
- Strings in switches
- Binary integral literals and underscores in numeric literals
- Multi-catch and more precise rethrow
- Improved Type Inference for Generic Instance Creation (diamond)
- Try-with-resources statement
- Simplified Varargs Method Invocation

Library Updates for CLDC

- Java NIO File I/O Subset
- Subset of NIO Buffers
  - StringBuilder and CharSequence, String formatter
- ServiceLoader API to support Service Providers
- Collections update
  - Collection, List, Set, Map
  - Implementations including Hashtable and Vector
  - Iterable and Iterator

Library Updates for CLDC – continued

- Subset of SE Logging API
- Comparable interface
- Closeable and AutoCloseable – Try with resources
- Annotations – SuppressWarnings, Deprecated, Override

Generic Connection Framework (GCF)

- Consolidated from CLDC 1.1, MIDP, CDC, JSR 197
- New Access Point API – WiFi AP, Network I/F, WAN, etc.
- New Connection Options
- Support for
  - HTTP, HTTPS,
  - Datagram, Multicast,
  - Socket, Secure Socket,
  - Serial Comm ports, etc.

Development Tools for CLDC 8

- Standard JDK tools are used for application development
  - Compiler
  - IDE – Eclipse / NetBeans
- ME SDK to target the application for Java ME platform
- Debug structures are optional
  - Debug features are ignored by the CLDC 8 product VM
  - Can be filtered out by tools at build time to save application footprint

Program Agenda

- Java ME 8 Platform
- CLDC Alignment with Java SE
- Java ME Embedded Profile
- Device Access API
- Java ME 8 Support for Mobile
- Looking Ahead to Java ME 9

Java ME Embedded Profile 8

Dedicated APIs for sub 10MB small embedded devices

- Core APIs - an evolution of IMP-NG (JSR 228)
  - App start, pause/resume, destroy, multitasking, system properties, etc
  - Dedicated Application Management System (AMS) API
  - Events model
  - Push Registry based inbound connection handling
  - Persistent storage
  - Allows diverse UI and media handler implementations
  - App access to cellular network properties on communication modules
- Simplifying and enhancing MIDP for M2M devices and communication modules

Embedded New APIs Compared to IMP-NG

- Inter-IMlet communication
- Shared libraries
- Notifications
- Auto Start applications
- Screen Saver and Idle Screen applications
- Support for Open Type fonts
- AMS API
- Power Management

Embedded New API

Text Only User Interface

- For simple devices with a line-oriented display only
- Key event support (in case a user input device is assigned)
- Scrolling support
- Support of single-line or multiple-line displays
- Support of more than one text color

Embedded New APIs

- Touch/Gesture API
- AMS API
  - Install, remove, update, start, stop, monitor,…
  - Trusted applications can use all AMS functions
  - All necessary callbacks provided
    - Status
    - Security
  - Useful to provide a customized application storefront or manage applications on the device

Services Framework

- Services Framework provides the ability for clients to use services that can be implemented by more than 1 service provider (SP)
- Service is a well known set of Interfaces/Abstract classes;
  Service Provider is an implementation of the Service
- Features of the Service Framework
  - Ability of a shared library to declare that it provides a service
  - Ability of a shared library or App to declare a dependency on a service
  - Ability of the AMS to bind a shared library provided by one App to another requesting a service
  - Enable the binding only if the shared library is trusted
  - Service provider executed in context of client application

Framework Design Components

- Main components of Services Framework
  - java.util.ServiceLoader provided in CLDC
  - Service provider interface defined by the service
  - Service provider implementation of the service interface
  - Support for consumers of services (client applications)
  - Support of Service Framework in AMS (install, update, delete)

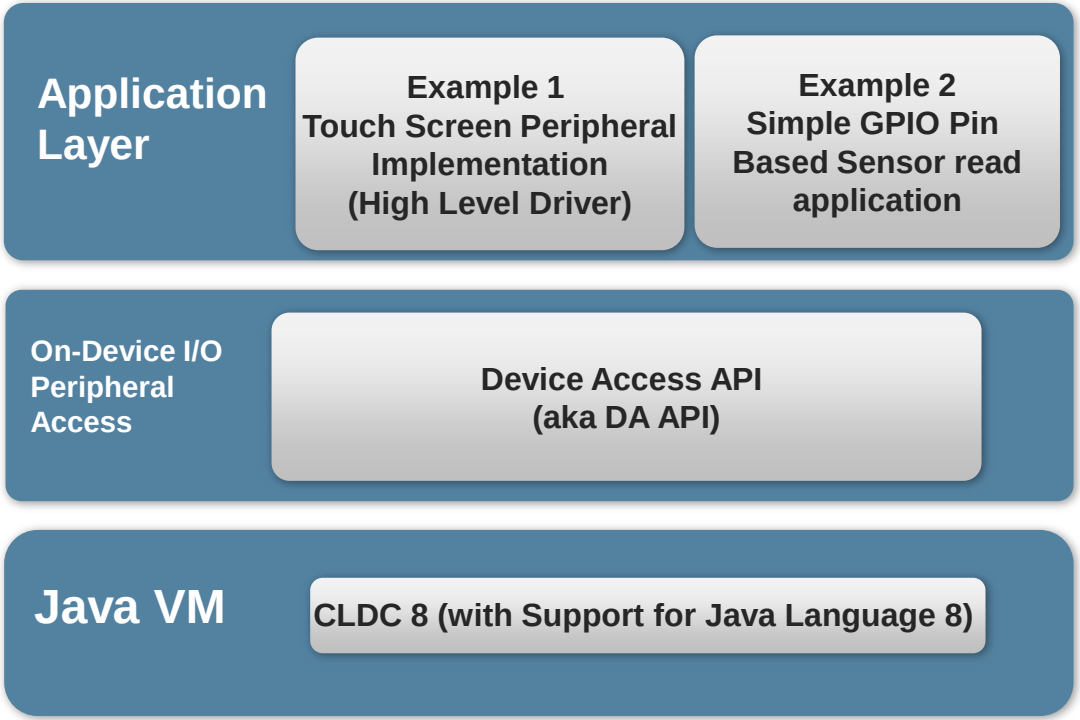
Program Agenda

- Java ME 8 Platform
- CLDC Alignment with Java SE
- Java ME Embedded Profile
- Device Access API
- Java ME 8 Support for Mobile
- Looking Ahead to Java ME 9

Device Access API

What is the DA API?

- Generic device peripheral access API
- Optional Package for small embedded devices
- Compatible with Java SE



Device Access API

What is a Peripheral in the DA API?

- Peripheral designates any kind of peripheral device, from a peripheral device external to the host device to a peripheral chip embedded in the host device.
- May designate a sub-component or a resource of that hardware peripheral device that can be accessed and controlled independently (to some extent) from the other components or resources of that same hardware peripheral device.
- May not always correspond to a single hardware peripheral device or component thereof, but may be composed of several other Peripherals.
  - For example, a Touch Screen Peripheral may be assembled from a Touch Screen SPI (Serial Peripheral Interface bus) slave Peripheral and a GPIO pin Peripheral handling the Pen-Down signal.

Device Access API

Device Type	Description
General Purpose Input/Output	GPIO pins as well as GPIO ports
Inter-Integrated Circuit Bus	I2C slave on the I2C bus
Serial Peripheral Interface Bus	SPI slave on the SPI bus
Analog To Digital Converter	ADC channels of an ADC converter
Universal Asynchronous Receiver/Transmitter	UART access and control
Memory-Mapped Input/Output	Devices with memory-mapped registers and memory blocks
AT Command Devices	MODEMs and in general devices supporting AT commands
Watchdog Timers	Watchdog access
Pulse Counter/Timer	Pulse counters and timers
Generic Devices	Other types of devices can be added

Program Agenda

- Java ME 8 Platform
- CLDC Alignment with Java SE
- Java ME Embedded Profile
- Device Access API
- Java ME 8 Support for Mobile
- Looking Ahead to Java ME 9

Mobile Support in Java ME Embedded 8

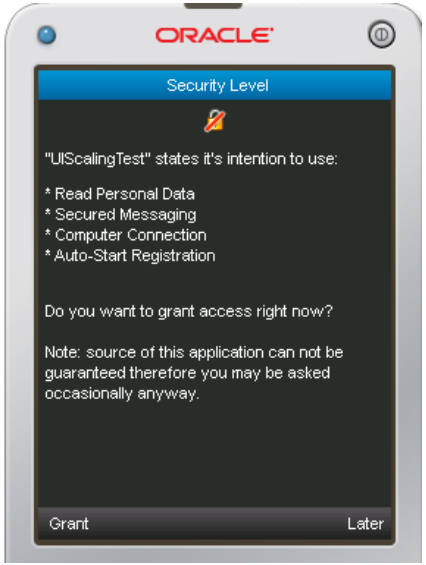
ME Embedded Profile support for feature phones

- Java ME 8 Embedded increases the reach of Java ME to a broader range of small connected devices – beyond feature phones
- Mobile developers will benefit from an expanded accessible market and access to a broader community and expertise
- Mobile OEMs can leverage the new features in ME Embedded Profile
- Java ME Embedded will offer building block options for OEMs to derive compact and performant mobile runtimes based on ME 8
- Oracle will encourage JCP standardization of mobile features by 3rd parties to address any outstanding requirements including 3rd party efforts to revise MSA
- Existing MSA JSRs will continue to be maintained in JCP

Mobile Support in Java ME Embedded 8

Simplifying and improving MIDP APIs

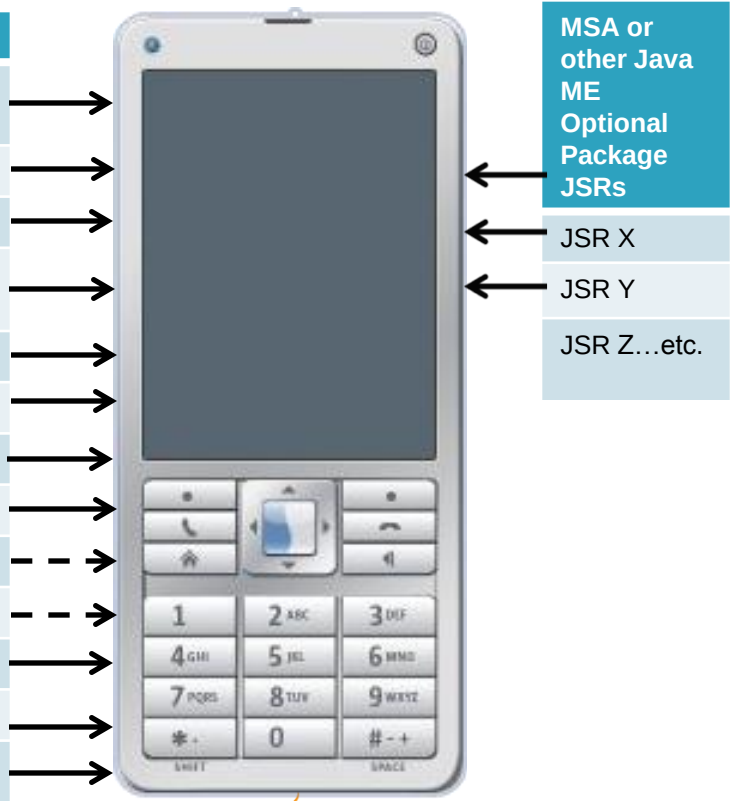
- Extensions
  - Shared Libraries
  - Screen Savers, Idle Screen widget
  - Multiple Displays, Notifications and Events
  - Gesture API, Higher-order pointer events
  - OpenType fonts packaged by application
- Simplified security model for improved user experience
  - Installation of signed unverified applications is allowed
  - Implementation is suggested to minimize the amount of user interventions related to security (security prompts)



“Mobile Vertical” support in Java ME Embedded 8

Example: Deriving a core mobile platform

Specification	Platform Functional Area
Java ME Embedded Profile 8	Application Packaging and Provisioning Management, Shared Libraries
	Cellular Network information (javax.microedition.cellular)
	System /Application Event subsystem (javax.microedition.event)
	Device Power Mode Control/Monitoring (javax.microedition.event.power)
	Inter-application communication and Push Registry
	Application execution model and concurrency
	Persistent data storage and retrieval (Full RMS)
	Security Policy and Application permissions
	(Optional) Graphical UI subsystem (javax.microedition.lcdui/lui)
	(Optional) Media manipulation (javax.microedition.media)
Device Access API	(Optional) UART and USB peripheral access
GCF 8	(Optional) Http, Https, Socket, UDP/Multicast, AccessPoint APIs
CLDC 8	Subset of Java SE 6/7/8 Language and Core APIs (lang, io, util, net, nio, security etc.)



Program Agenda

- Java ME 8 Platform
- CLDC Alignment with Java SE
- Java ME Embedded Profile
- Device Access API
- Java ME Java ME 8 Support for Mobile
- Looking Ahead to Java ME 9

Java ME 9

Directions

- ME / SE architectural alignment
  - Introduce SE 9 Modularity to mobile/embedded
- Upper stack separated from the Core VM
  - Relevant Mobile and Embedded APIs can run or ME or SE
  - ME vs. SE becomes a footprint/functionality tradeoff
- New APIs for small embedded devices
- Mobile support from Java SE development and management tools

Java ME 9

Example – Location API

- Identify use cases for Location across Java Platform
  - On tablets, on desktop, servers, etc.
- Explore update to JSR 179/293 or a new JSR to support Java SE
  - Spec, RI for SE, TCK support
- Potential updates for:
  - Location provider for getting location based on IP address lookup
  - Add an API for accelerometer so only one API is needed for navigation using compass and accelerometer
  - Modularize the API to fit with Java SE 9 module system

JCP Activities for Java ME 8

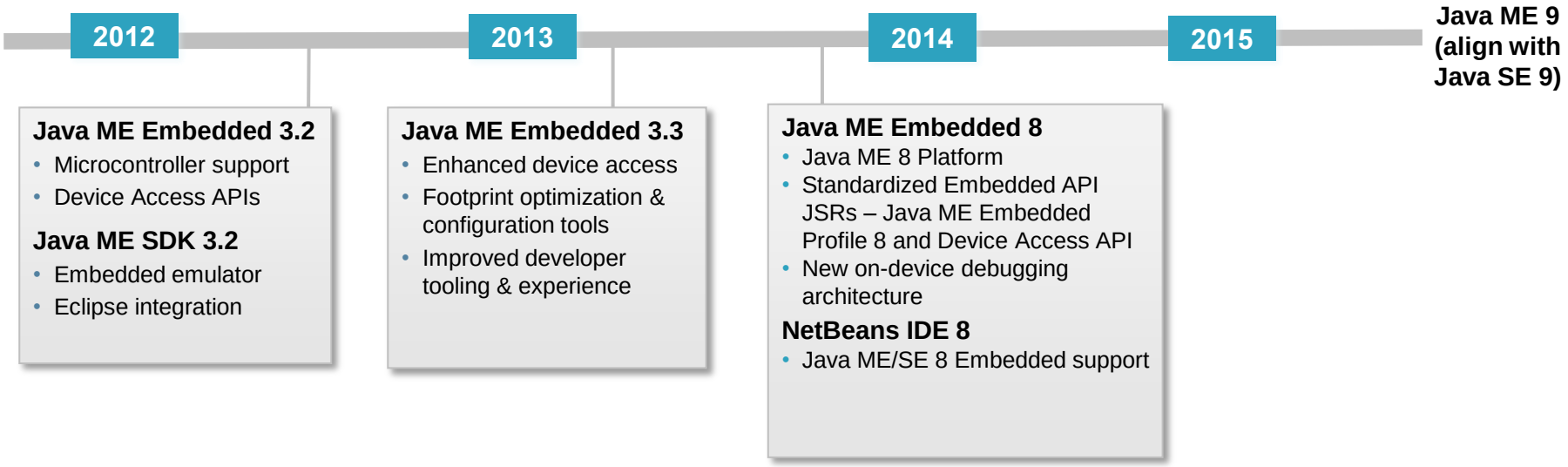
- Connected Limited Device Configuration 8 JSR Proposal – JSR 360
  - Initial EG members Aplix, Nokia, North Sixty-One, Stefano Andreani
  - Supporters: ARM, Deutsche Telekom, Siemens, TOTVS, Vodafone
- Java ME Embedded Profile – JSR 361
  - Initial EG members: Aplix Corp., Cinterion, Deutsche Telekom, North Sixty-One Ltd, Stefano Andreani
  - Supporters: ARM, Siemens, TOTVS, Vodafone
- No mobile specific JSRs (MSA/MIDP) filed by Oracle
  - Will encourage 3rd party efforts to revise MSA
- JCP completion within a year

Summary

The Java ME 8 Platform Release…

- Creates a new platform definition based on needs of the small embedded device market and its services environments
- Initiates Java ME-SE API alignment, a unified ecosystem view and expanded opportunities for Java SE and Java ME developers and device OEMs
- Represents a significant standards driven modernization of the Java ME Platform and continuing support for a low-mid tier core mobile phone device platform
- Two JSRs submitted - CLDC 8 and Java ME Embedded Profile

Java ME Roadmap



Related Sessions and BOFs

- CON7110: Developing Java Mobile and Embedded Applications with Java ME SDK 3.2
  Tuesday, Oct 2, 11:30 AM - Hotel Nikko - Nikko Ballroom II/III
- CON3984: IMP – A Profile for an Embedded World with Increasing Demands
  Wednesday, Oct 3, 3:00 PM - Hotel Nikko - Nikko Ballroom II/III
- Presented on Monday:
  - CON4247: CLDC: The Java Platform for Feature Phones and Low-Footprint Embedded Devices
  - CON5943: Java ME Service Platform
  - CON5804 - A New Platform for Ubiquitous Computing: Oracle Java ME Embedded

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.


JavaOne™