



Java is a trademark of Sun Microsystems, Inc.

# JavaOne<sup>SM</sup>

Hardware Accelerated  
Graphics for High  
Performance JavaFX™  
Mobile Applications

Pavel Petrosenko, Sun Microsystems  
Jan Valenta, Sun Microsystems  
Jerry Evans, Sun Microsystems

# Goal of this Session

Demonstrate high performance graphics capabilities of JavaME™ and JavaFX™ Mobile Platforms.

Describe how we optimized the platform to leverage hardware acceleration.

Demonstrate how platform optimizations help to run compelling graphical applications on the platform.

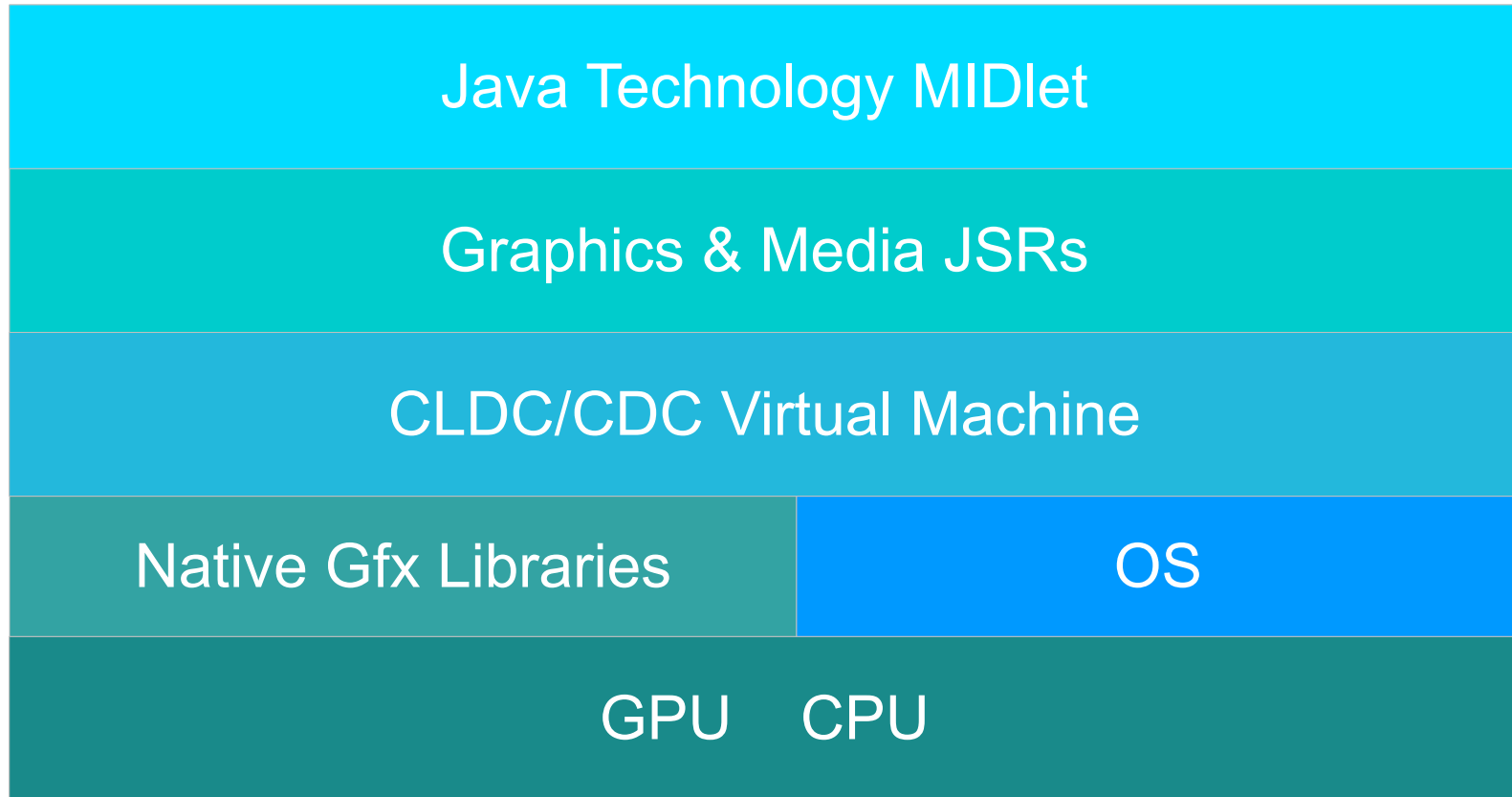
# Agenda

- > JavaME™ Platform overview
- > Graphics acceleration APIs
- > HW acceleration on Java Mobile platform
- > JavaFX™ Mobile Platform overview
- > JavaFX™ Mobile Platform performance
- > References

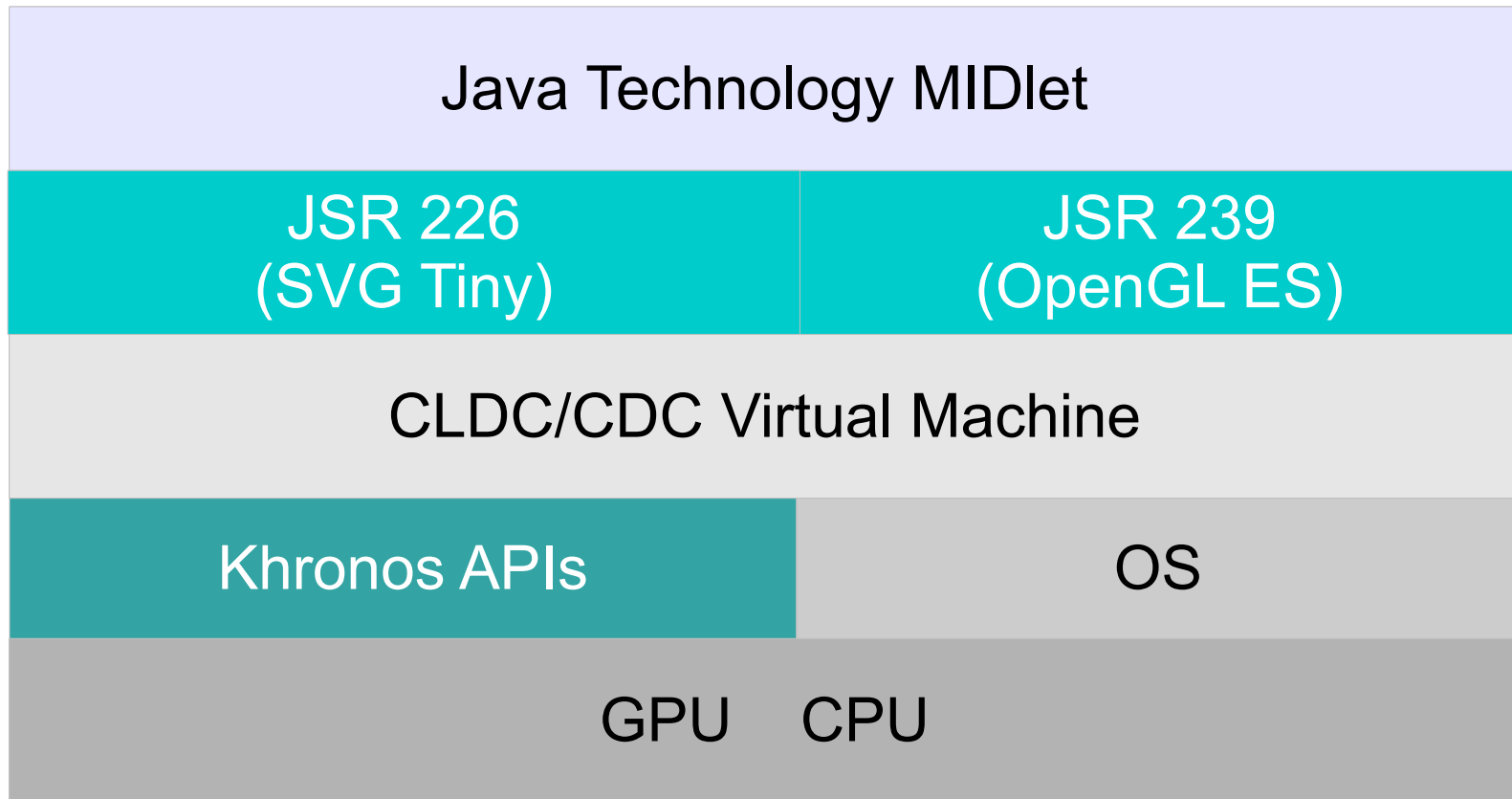
# Why accelerate?

- > Reach content developers
  - More features/higher applications complexity
  - Richer UI and graphics content
  - Interactivity
- > Device characteristics
  - Larger screen sizes/higher pixel density
  - Limited CPU capabilities

# JavaME™ Platform Overview



# JavaME™ Platform implementation



## JSR 226

- > Scalable 2D Vector Graphics API for Java Micro Edition (JavaME™ platform)
  - Java API to draw, manipulate and animate SVG Tiny 1.1 content
  - Part of Mobile Service Architecture (JSR 248) – wide deployment

## JSR 239

- > JSR 239 defines Java bindings to the OpenGL<sup>®</sup> ES 1.0 and 1.1
- > Immediate mode API
  - Highly flexible control of rendering
- > Low level hardware oriented 3D API
- > Access to latest hardware features



# Khronos Media Acceleration APIs

- > Khronos APIs
  - Royalty-free, cross-platform
  - Supported by a large number of hardware vendors for mobile devices
- > OpenGL ES - 3D graphics on embedded systems
- > OpenVG – Low-level API for 2D vector graphics
- > EGL – interface between Khronos rendering APIs and the underlying native platform
- > OpenMAX - Streaming media acceleration

# OpenGL ES



- > 2D and 3D graphics on embedded systems
  - Subset of desktop OpenGL
  - OpenGL ES - specific extensions
- > OpenGL ES 1.x
  - For fixed-function hardware
- > OpenGL ES 2.x
  - For programmable hardware

# OpenVG

- > OpenVG 1.0
  - Low level graphics API specially designed for 2D vector graphics (SVG, Flash)
  - Support for raster graphics as well
- > OpenVG 1.1
  - Adds accelerated text rendering and multi-sampled antialiasing
- > Can be implemented on top of OpenGL ES implementation
- > Wide variety of applications

# Coprocessors & Hardware Accelerators

- > Support of graphics/video accelerators is essential for gaming experience
- > JIT Compiler needs to be aware of JSR implementation
- > Floating-point coprocessors
  - ARM VFP
- > ARM Jazelle DBX makes cold startup faster
- > Instruction set extensions (XScale, Jazelle RCT)

# JSR 226 Demo

- > TI OMAP 3440 (Zoom MDK)
- > SVG content courtesy of <http://www.tinyline.com>

# JSR 239 Demo

- > JBenchmark239 application
  - <http://www.jbenchmark.com>
- > Running on a development phone

# JavaFX, JavaFX Mobile

- > The language
  - Bindings, declarations, sequences, interpolation
  - Simplified content authoring
- > Desktop, Browser, Mobile, TV, ...
  - Common vs. Desktop Profile (Swing, advanced visual effects)
- > Tools support
  - IDE plugins, designer suites
- > Full access to Java classes

## JavaFX Mobile

- > Brings expressive, rich user interface to mobile world
- > Helps to solve the fragmentation issue
- > Easy:
  - Development of GUI thanks to the language and tools,
  - Access to phone features thanks to Java ME
- > Are today's phone powerful enough?
  - Yes, but there is never enough performance
  - HW acceleration is important



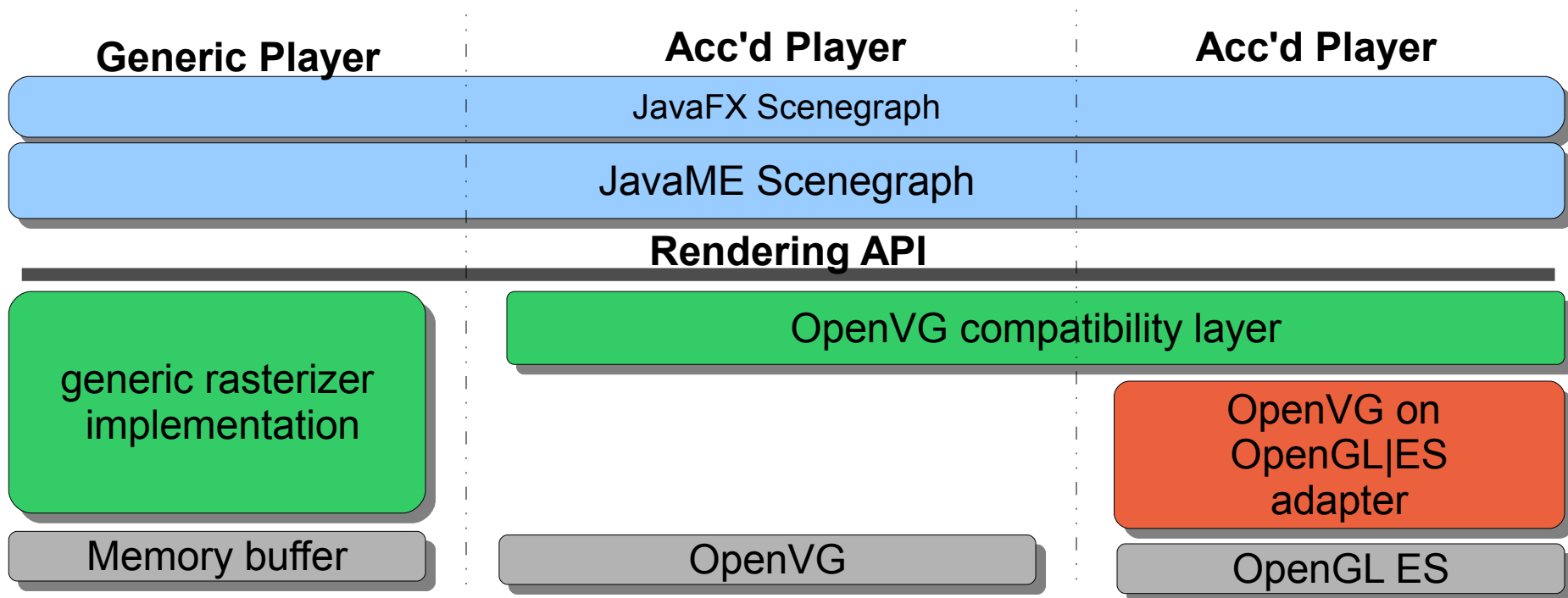
# JavaFX Mobile in Java ME SDK 3.0

- > Based on Netbeans platform
- > Works on Windows, soon on Linux and MacOS
- > Mobile emulator skins with JavaFX Mobile support
- > Allows profiling JavaFX Mobile applications



# FX Players Architectures

We focus on accelerated players



# Rendering API

- > General graphical low level Java API
  - Based on API used in PhoneME
  - Path, stroke, fills (radial and linear gradients, solid color), images, textures, transformations, clipping
  - Antialiasing and transparency support
  - Rendering model similar to SVG, Flash
- > Implementations:
  - Software renderer: memory surface (multiple types), portable, C language
  - OpenVG acceleration
  - OpenGL|ES acceleration

# Scenegraphs

- > Layers between JavaFX Script and the renderer
- > Keeps track of the scene structure, both visually and logically. Tree graph of the scene.
- > JavaFX Scenegraph
  - Interfaces JavaFX and Java
  - Preserves scene hierarchy, remembers nodes
  - `javafx.scene.Scene` class
- > Java Scenegraph
  - Level of shapes, images, paths
  - Nodes, state, animation, effects

# JavaFX™ Demo#1

- > HTC Touch Diamond
  - OpenGL | ES based HW acceleration
  - Powered by AmanithVG
  - MediaStore

# JavaFX™ Demos#2

- > TI OMAP 3430 Zoom MDK
  - OpenVG based HW acceleration
  - Image benchmark
    - 15 fps vs. 45 fps
  - FishSim
    - 0.6 fps vs. 15 fps, 99% CPU vs. 15%CPU

# JavaFX™ Demo#3

- > HTC Touch Diamond
  - FishSim side by side
  - Powered by AmanithVG

## For More Information

- > PhoneME - <http://phoneme.dev.java.net/>
- > JavaFX™ - <http://javafx.com>
- > JavaFX™ Mobile -  
<http://www.sun.com/software/javafx/mobile/>
- > Java ME SDK 3.0 -  
<http://java.sun.com/javame/downloads/sdk30.jsp>
- > Khronos APIs - <http://www.khronos.org>
- > TI OMAP Zoom MDK page - <http://www.omapzoom.org>
- > AmanithVG - <http://www.amanithvg.com>





# JavaOne<sup>SM</sup>

# Thank You

Pavel Petrosenko (pavel.petrosenko@sun.com)  
Jan Valenta (jan.valenta@sun.com)  
Jerry Evans (jerry.evans@sun.com)