Mentor Graphics
EMBEDDED SYSTEMS

Getting Started with Android Development for Embedded Systems

Colin Walls

colin_walls@mentor.com

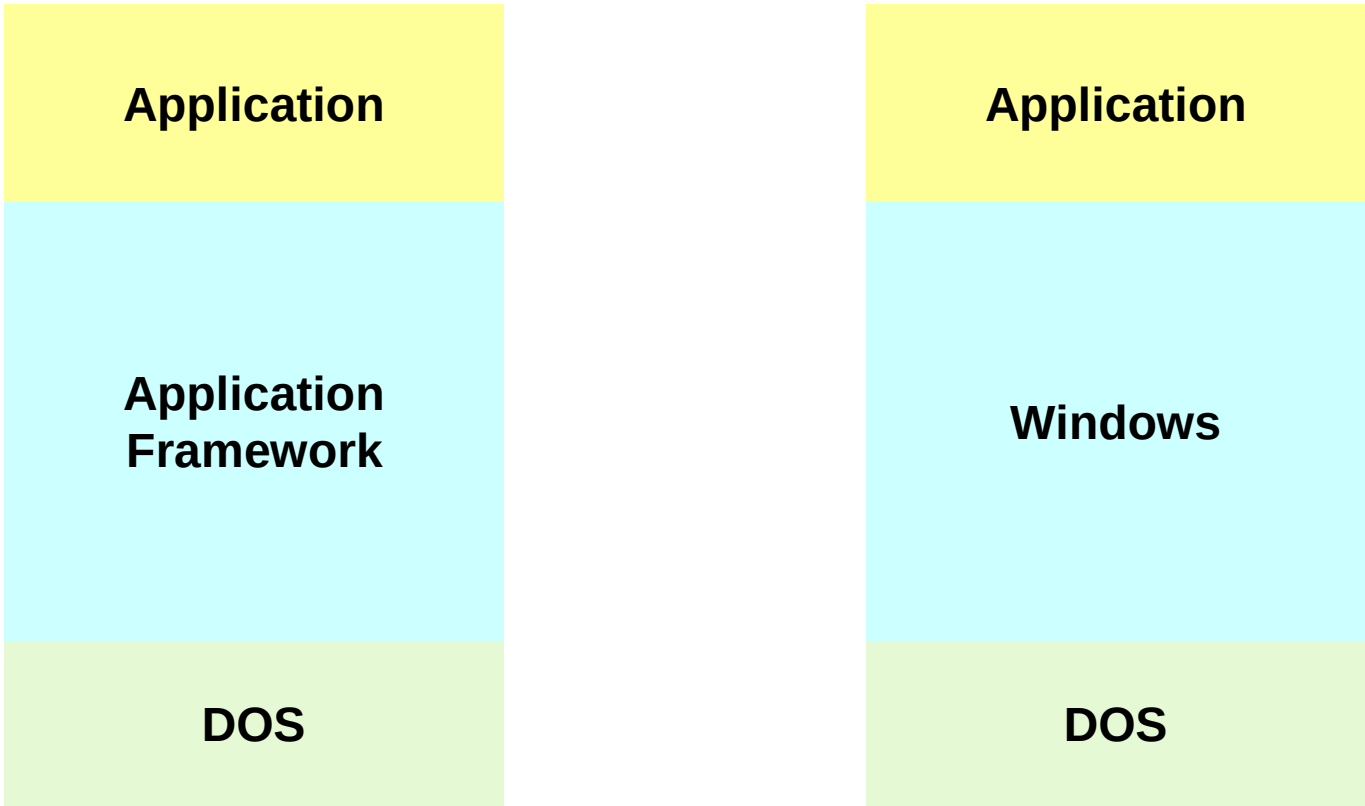
Agenda

- Introduction
- Android architecture
- Application development
- Android deployment
- Mentor Graphics support for Android
- Conclusions

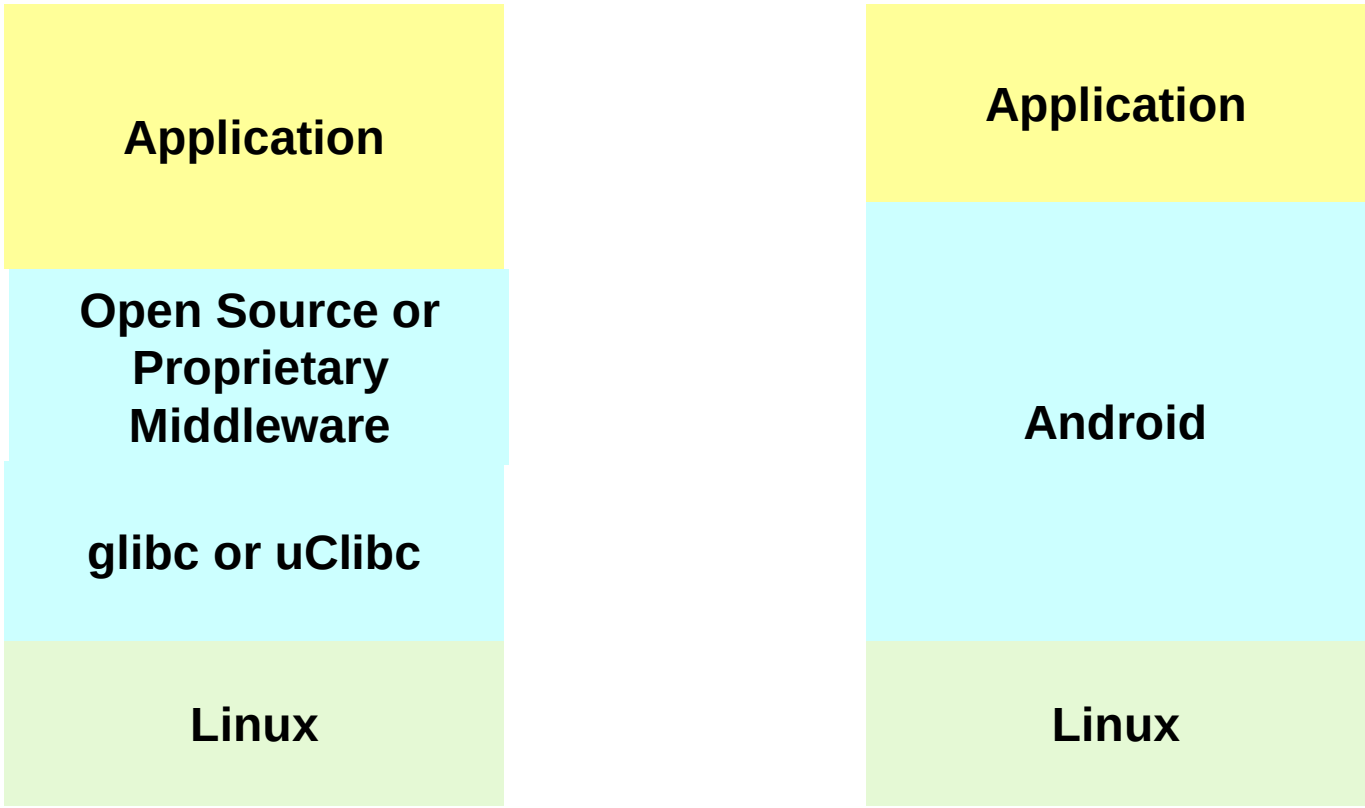
What is Android?

- An operating system for smart phones?
  - yes and no
- An integrated software platform for building smart phones?
  - almost
- An integrated software platform for building connected devices
  - application framework on top of Linux
  - open source – not GPL

What is Android?



What is Android?



Android History

2005	Google acquires Android Inc.
2007	Open Handset Alliance announces Android
2008	First Android based handset launched
2008	Android source code released
2009	Mentor Graphics acquires Embedded Alley

Agenda

- Introduction
- **Android architecture**
- Application development
- Android deployment
- Mentor Graphics support for Android
- Conclusions

Android Architecture

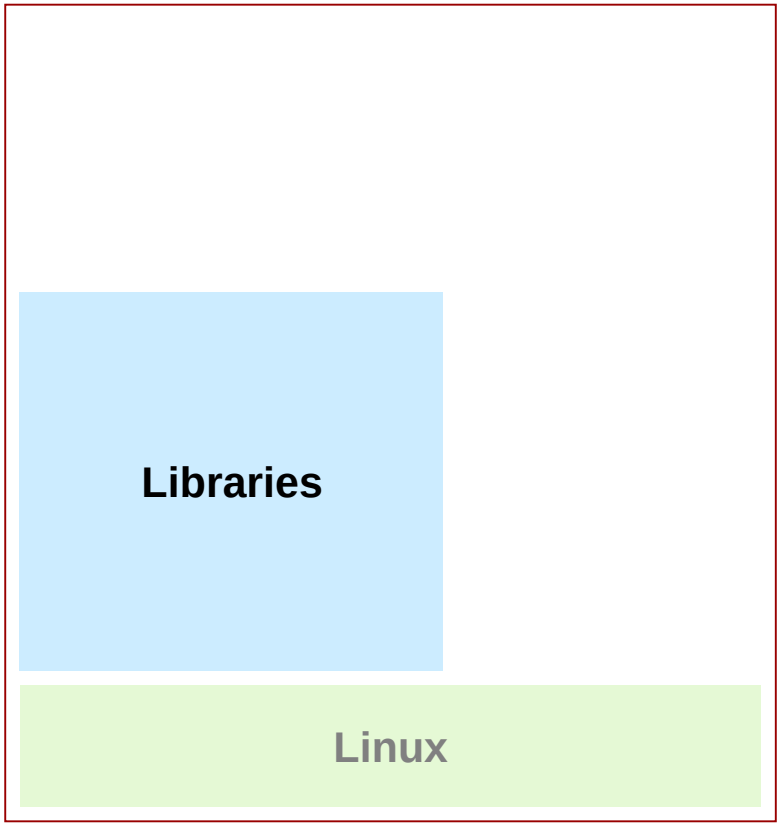
Linux

- 2.6.3x
  - 115 patches required
- provides:
  - process management
  - memory management
  - security
  - networking
  - drivers:
    - display, keypad, camera, WiFi, flash, audio, IPC, power

Android Architecture

Libraries

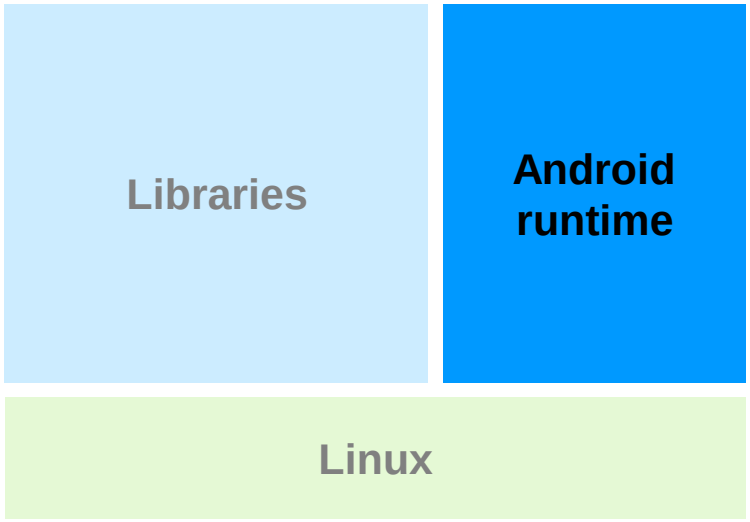
- e.g.:
  - **`libc`** - **`bionic`**
  - media libraries
  - graphics
  - SQLite



Android Architecture

Android runtime

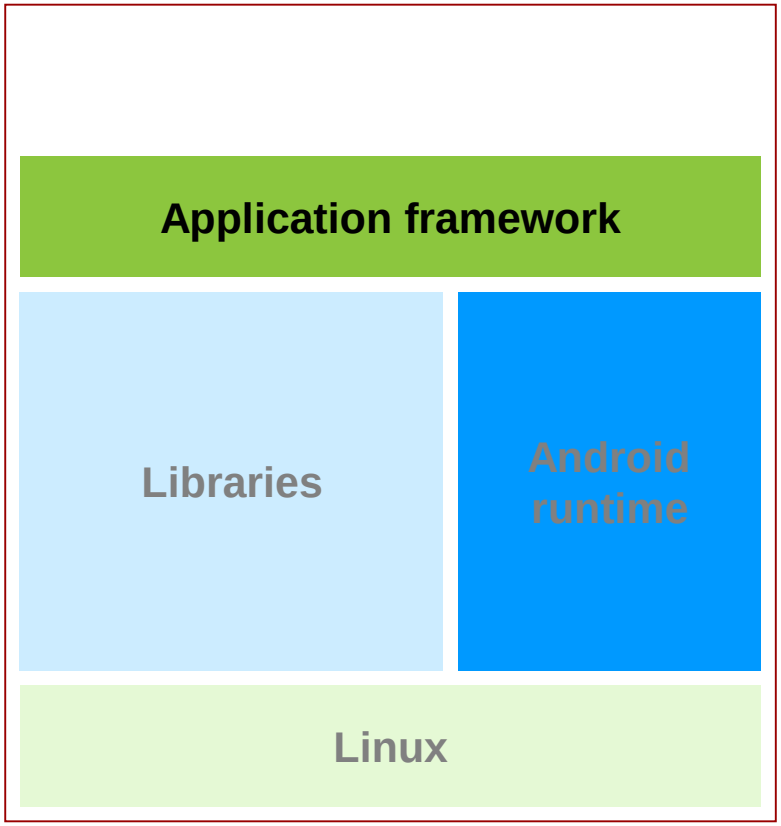
- Dalvik VM
- Not a Java VM
  - register based
- one instance per application
- memory optimized
- uses Linux to manage memory and multi-threading



Android Architecture

Application framework

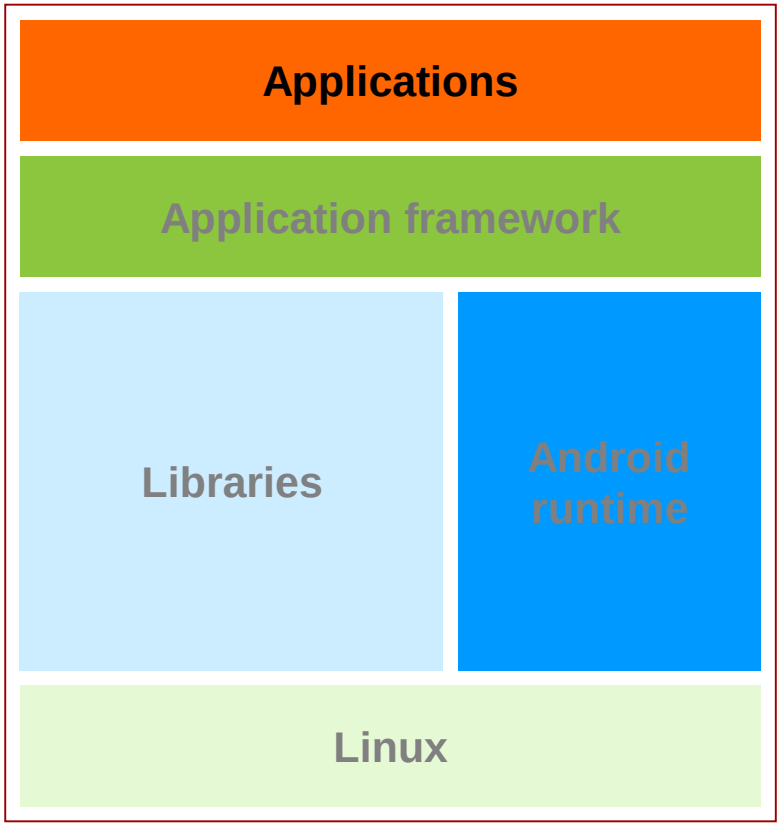
- services and systems:
  - views
  - content providers
  - resource manager
  - notification manager
  - activity manager
- all Java classes
- any application can publish its capabilities



Android Architecture

Applications

- programs provided:
  - email
  - SMS
  - calendar
  - Web browser
  - contacts
- all applications have the same status
- Java programs
  - Uses standard Java tools
  - Converted to Dalvik bytecode



Agenda

- Introduction
- Android architecture
- **Application development**
- Android deployment
- Mentor Graphics support for Android
- Conclusions

Development Environment

- Eclipse based
  - Android Development Tools (ADT) plug-in
  - Create an Android Virtual Device (AVD)
  - Android Emulator

    or
  - Target device
    - T-Mobile G1
    - Android Dev Phone 1
    - USB connection
  - Only ARM devices

Programming Model

- All the resources for an application are bundled into an archive – Android package
- Programs written in Java
  - run using Dalvik VM
  - use non-standard, memory efficient bytecodes
  - each program has its own Linux process
  - data protected, but sharing possible

Programming Model

No single entry point – **`main()`**

- components instantiated and run as required
- types of components:
  - activities
  - services
  - broadcast receivers
  - content providers

Programming Model

Activities

- executable unit that performs specific function
- has UI
- application may have numerous activities
- one activity nominated as default [entry point]

Services

- similar to activity
- no UI
- runs in the background

Broadcast receivers

- respond to broadcast messages

Content providers

- makes application's data available to others
- can be:
  - file system
  - SQLite
  - other
- targeted by **`ContentResolver`**

Programming Model

Manifest file

- describes components of an application
- structured XML file
  - called **`AndroidManifest.xml`**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest . . . >
    <application . . . >
        <activity android:name="com.example.project.MyActivity“
        . . . >
        </activity>
        . . .
    </application>
</manifest>
```

Programming Model

Intents

- asynchronous messages
- activate activities, services, and broadcast receivers
- for activities and services, specifies action and location of data
- applications have intent filters

Programming Model

Intent filters

- describe what intent the components of an application can handle
- in manifest file

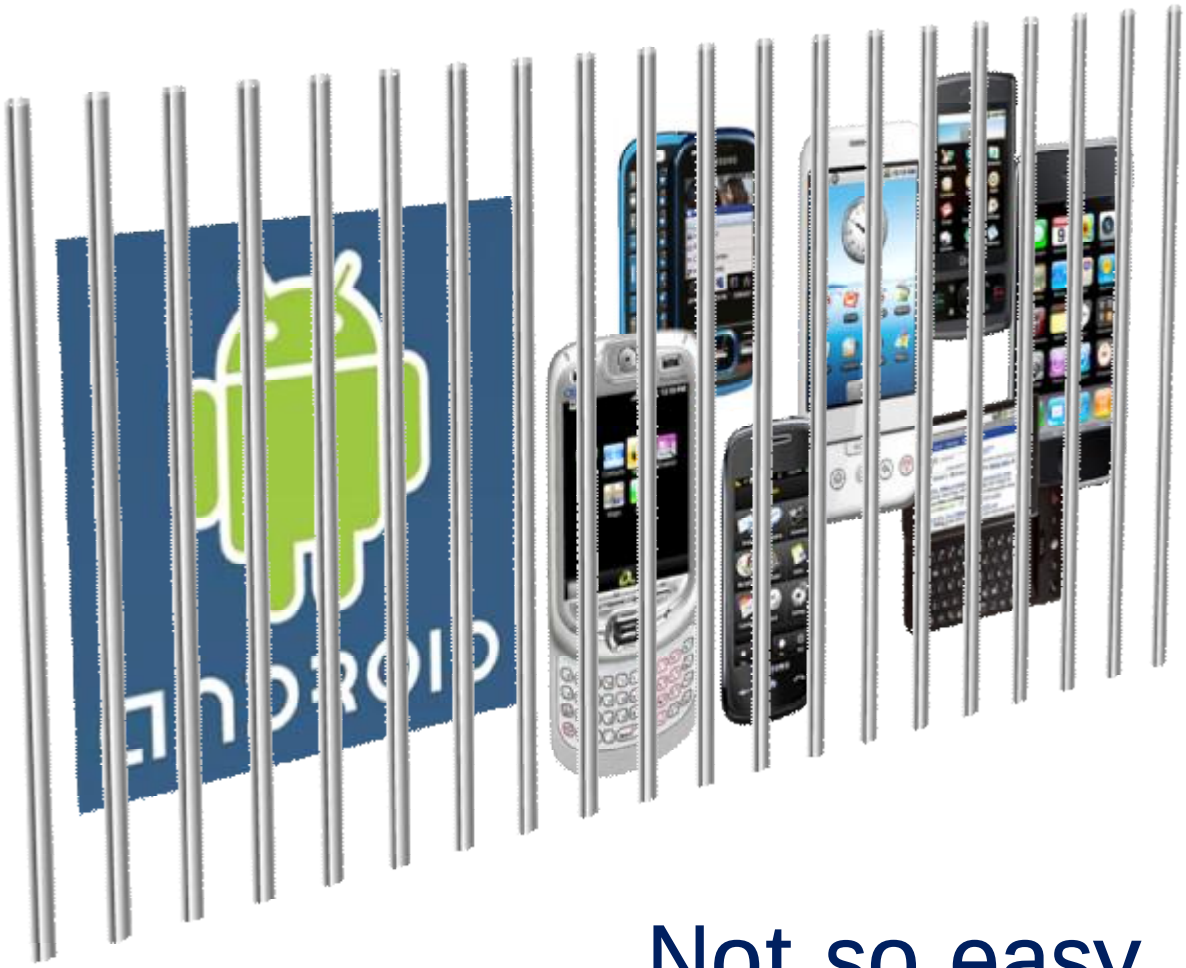
```
<application . . . >
    <activity android:name="com.example.project.MyActivity“
    . . . >
        <intent-filter . . . >
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    . . .
</application>
```

Agenda

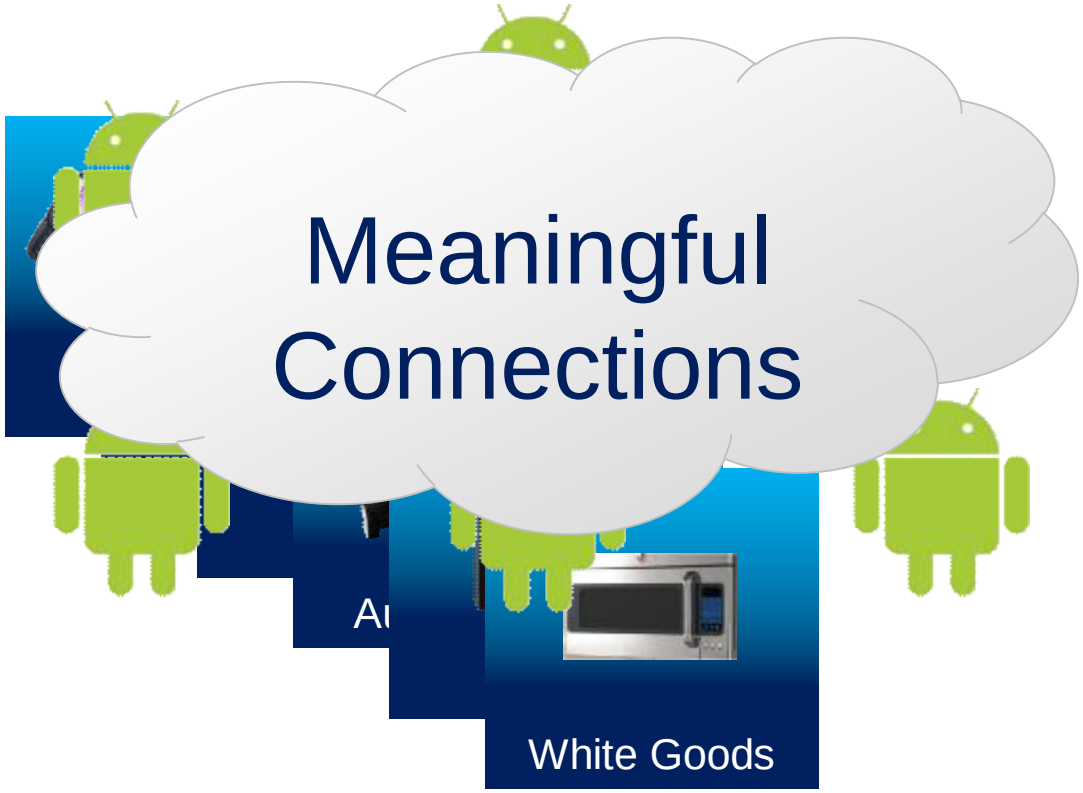
- Introduction
- Android architecture
- Application development
- **Android deployment**
- Mentor Graphics support for Android
- Conclusions

Android is Designed for Mobile Phones

Android Experience on Any Platform?



Not so easy...



Mentor Graphics
EMBEDDED SYSTEMS

Medical

Automotive

White Goods

Telecom

Mentor Graphics
EMBEDDED SYSTEMS

Agenda

- Introduction
- Android architecture
- Application development
- Android deployment
- **Mentor Graphics support for Android**
- Conclusions

THE BEST WAY TO BUILD WITH OPEN SOURCE

The Embedded Alley approach

Offer software and tools that precisely fit the customer's product platform

Maximize Open Source leverage for customers

Collaborate closely throughout the customer product life cycle

Bridging the Smartphone Gap

- Today Android strongly tied to 3G mobile devices
  - Chipsets, development tools, ecosystem (ARM, Qualcomm)
  - Platform developer community anchored in Open Handset Alliance
- Needs investment to enable wider market
  - Embedded CPUs, device drivers, board support
  - Broader open source community
- Embedded Alley is bridging that gap
  - Enabling popular embedded CPUs and chipsets
  - Extending reach of Android
  - Runtime porting/optimizations, development tools

Embedded Alley and Android

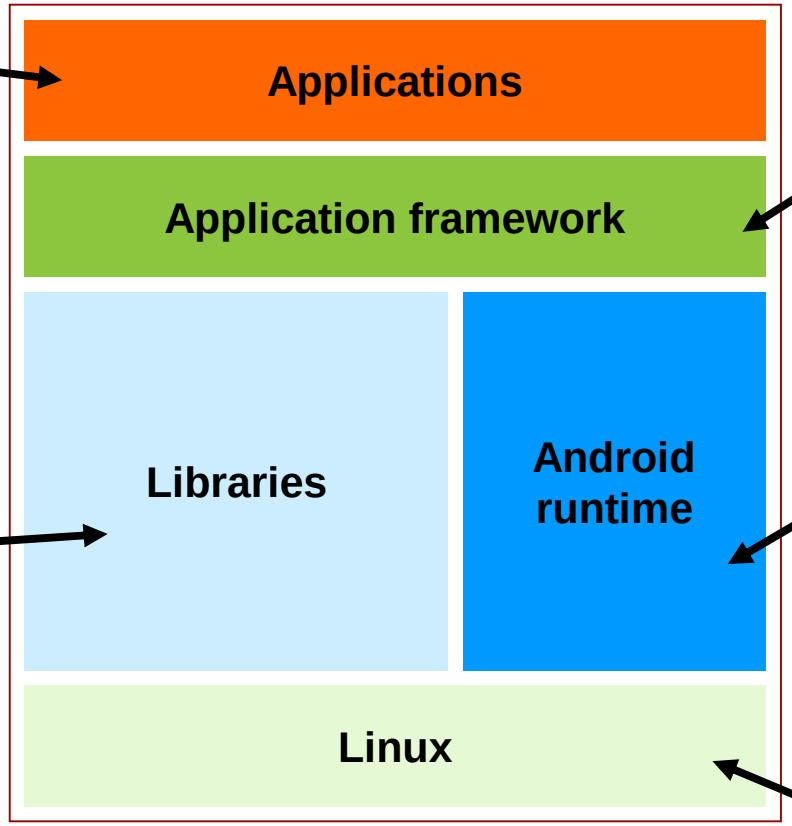
- Dramatically expands Android application space
  - Handheld multimedia
  - Digital video
  - Automotive
  - Medical devices
  - Home automation and SOHO networking
  - Instrumentation and industrial control






Android Deployment

Enable Customers to build product specific applications, UI
Produce product specific SDK and emulator



Extending beyond handset assumptions
Enable developing product and industry specific classes

Porting and optimizing Android Dalvik VM for CPU cores and SoCs

Porting/optimizing for instruction sets
Integrating with hardware acceleration
Extending beyond handset assumptions
Enable product and industry specific middleware

Providing Android-ready Linux kernel
Integrating and testing board support and industry-specific device drivers

Development System for Android

- Provides the latest stable Android runtime (Cupcake, 1.5)
  - Includes optimized Libraries for Architecture and CPU core
  - Includes optimized Dalvik VM for the Architecture and the CPU core
- Platform – Libraries, Kernel
  - Developing and adding board ports and custom drivers
  - Developing and adding product specific C/C++ Libraries (more than NDK)
  - Enables customizing HAL for hardware specific needs
- Runtime – Dalvik, Java Native Interface
  - Developing and adding product specific JNI
- Application Framework
  - Developing and adding product specific classes
  - Enables customizing the UI, theme, and icons
- Application SDK and Emulator (modified ADT, JDT Eclipse plug in)
  - Create product specific Application SDK
  - Enables customizing the emulator to match the product (hardware, behavior and skins)

Embedded Alley Development System

- Eclipse IDE for application development, tools and debug
  - Latest stable gcc, binutils and gdb/gdbserver (CDT)
  - Dnload app, console window, automatic debug connection (DSDP)
- Match Customer requirements and workflow
  - Integration and build tool to create product specific runtime
  - Build both proprietary code and open source components
  - Control build type and flags - debug, code coverage, optimization levels, CPU flags
  - Integrate with customer workflow and source control
- Best practices and real world workflow
  - Reproducible builds
  - Software BoM, Source package, License Auditing and Traceability
  - Multiple build targets - development, test, production, manufacturing
- User documentation

Android Services

- Custom board ports – kernel and Android
- Extend Android I/O support with JNI and Class support
- Integrate product/industry specific middleware
- Turnkey solutions
- Application development
- Integration and Support services

Agenda

- Introduction
- Android architecture
- Application development
- Android deployment
- Mentor Graphics support for Android
- **Conclusions**

Conclusions

- Android is a disruptive technology
  - introduced for mobile phones
  - much wider potential
- Many challenges in wider Android deployment
- Mentor Graphics is committed to supporting developers who choose Android/Linux

Mentor Graphics®
EMBEDDED SYSTEMS

Questions?

Colin Walls

colin_walls@mentor.com

blogs.mentor.com/colinwalls