Mobility Programming

Lecture 2: Android Architecture

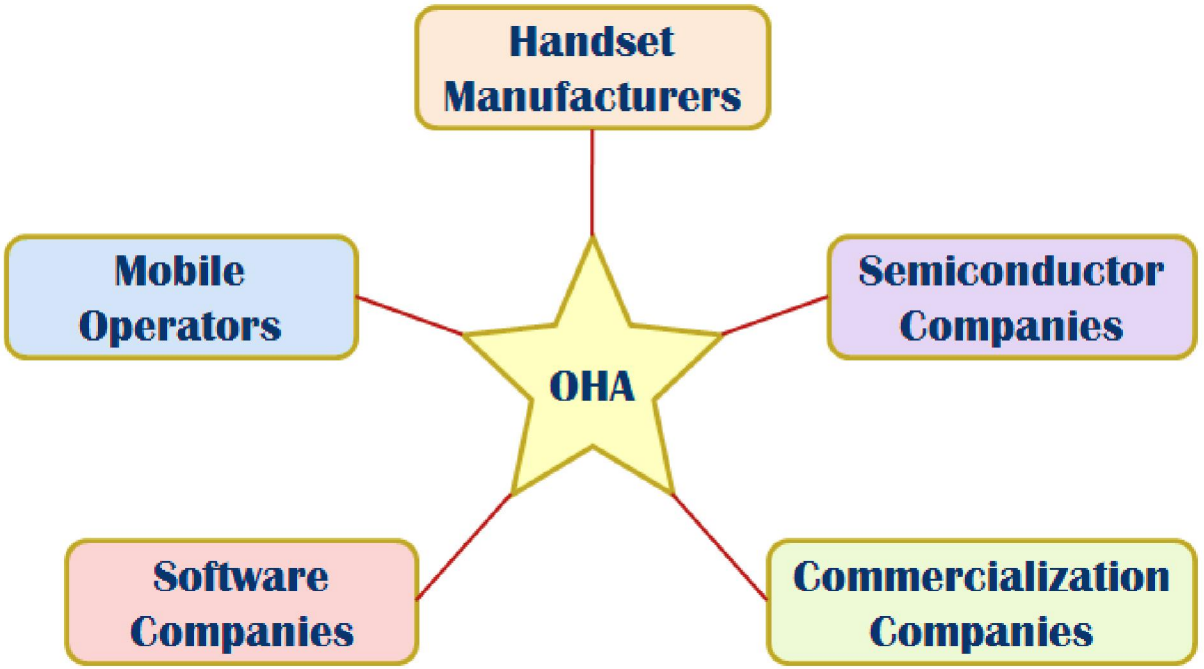
DR. RAMEZ ALKHATIB

What is Android?

- Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

OHA (Open Handset Alliance)

- A business alliance consisting of 47 companies to develop open standards for mobile devices



Android

- Android, Inc. founded in Palo Alto, California in October 2003
- Google acquired Android Inc. in August 2005
- Developed a mobile device platform powered by the Linux kernel
- Google marketed the platform to handset makers and carriers on the premise of providing a flexible, upgradable system
- On November 2007, the Open Handset Alliance, a consortium of several companies (e.g., Broadcom, Google, HTC, Intel, etc. unveiled itself). The goal is to develop open standards for mobile devices.
- Open Handset Alliance unveiled their first product, Android, a mobile device platform built on the Linux kernel version 2.6
- Android OS (open source) released in October 2008

Why Android

- Simple and powerful SDK
- No licensing fees
- Excellent documentation, and a thriving developer community
- From commercial perspective
  - Requires no certification for becoming an Android developer
  - Provides the Android Market for distribution and monetization of your application
  - Has no approval process for application distribution
  - Gives you total control over your brand and access to the user's home screen

Android Overview

- Open source OS and development platform
  - In theory, you can change anything
  - In practice....
- Hardware reference design
- Linux OS kernel
- Open-source libraries for app development
  - E.g., SQLite, Webkit, OpenGL, media managerMobile Application Design and D
- SDK and tools
- Preinstalled apps
- Wild west of app stores: the Market

Android Version

Code name	Version number	Initial release date
(No codename)[2]	1.0	September 23, 2008
(Internally known as "Petit Four")[2]	1.1	February 9, 2009
Cupcake	1.5	April 27, 2009
Donut[3]	1.6	September 15, 2009
Eclair[4]	2.0 – 2.1	October 26, 2009
Froyo[5]	2.2 – 2.2.3	May 20, 2010
Gingerbread[6]	2.3 – 2.3.7	December 6, 2010
Honeycomb[7]	3.0 – 3.2.6	February 22, 2011
Ice Cream Sandwich[8]	4.0 – 4.0.4	October 18, 2011
Jelly Bean[9]	4.1 – 4.3.1	July 9, 2012
KitKat[10]	4.4 – 4.4.4	October 31, 2013
Lollipop[12]	5.0 – 5.1.1	November 12, 2014
Marshmallow[13]	6.0 – 6.0.1	October 5, 2015
Nougat[14]	7.0 – 7.1.2	August 22, 2016
Oreo[15]	**8.0 – 8.1**	August 21, 2017
Android P	9	

Features and Specifications

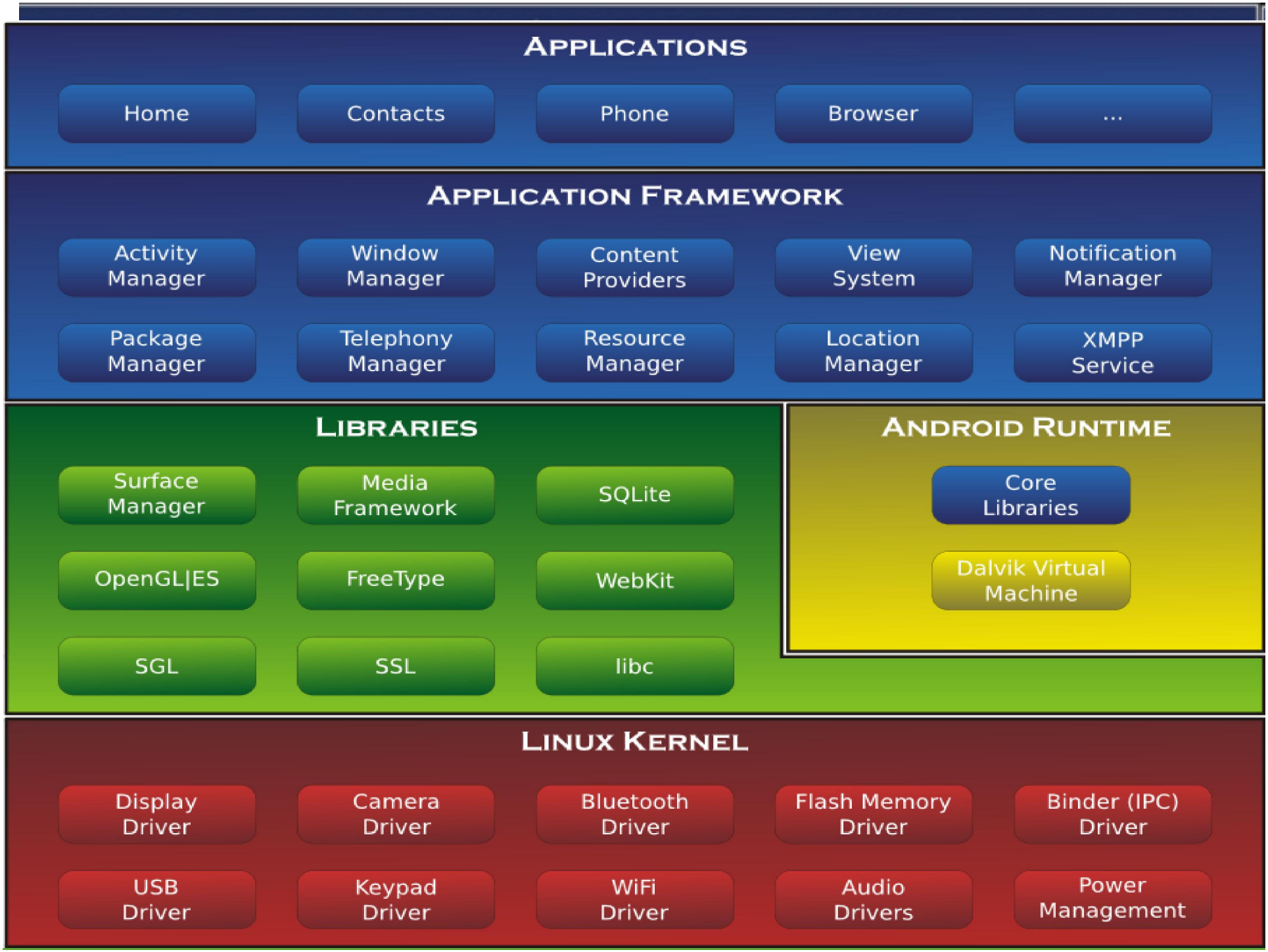
- **Platform** is adaptable to larger, VGA, 2D graphics library, 3D OpenGL graphics library
- **Storage** - SQLite, a lightweight relational database
- **Connectivity** - supports connectivity technologies including GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
- **Messaging** – SMS, MMS, threaded text messaging, Push Messaging service.
- **Multiple language support**
- **Web browser** - based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine.
- **Java support** – no Java Virtual Machine, Dalvik executables and run on Dalvik

Features and Specifications

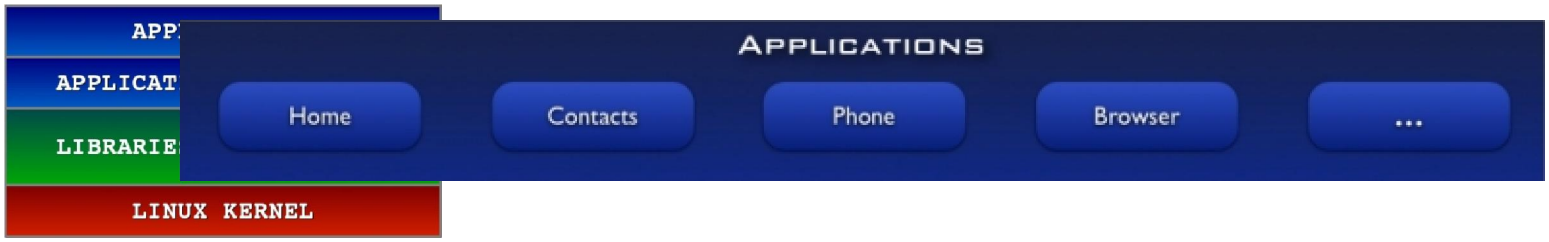
- **Media support** - audio/video/still media formats: WebM, H.263, H.264, MPEG-4 SP, WAV, JPEG, PNG, GIF, BMP, etc.
- **Streaming media support** - RTP/RTSP streaming (3GPP PSS, ISMA), HTML5 <video> tag, Adobe Flash Streaming (RTMP), HTTP Dynamic Streaming, Apple HTTP Live Streaming
- **Additional hardware support** - video/still cameras, touchscreens, GPS, accelerometers, gyroscopes, magnetometers, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D bit blits and accelerated 3D graphics
- **Multi-touch**
- **Bluetooth** - supports A2DP, AVRCP, sending files (OPP), accessing the phone book (PBAP), voice dialing and sending contacts between phones. Keyboard, mouse and joystick (HID)

Features and Specifications

- **Video calling** – no native video calling, but some handsets have a customized version of the operating system that supports it. Video calling through Google Talk is available in Android 2.3.4 and later. Skype 2.1 offers video calling in Android 2.3, including front camera support.
- **Multitasking**
- **Voice based features** - Google search through voice and voice actions for calling, texting, navigation, etc.
- **Tethering**


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
XMPP Service
Libraries
Surface Manager
Media Framework
SQLite
OpenGL|ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Bluetooth Driver
Flash Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Android S/W Stack - Application



- Android provides a set of core applications:
  - mail Client
  - SMS Program
  - Calendar
  - Maps
  - Browser
  - Contacts
  - Etc
- All applications are written using the Java language.

Android S/W Stack – App Framework

Enabling and simplifying the reuse of components

- ✓ Developers have full access to the same framework APIs used by the core applications.
- ✓ Users are allowed to replace components.

Android S/W Stack –App Framework (Cont)

Features

Feature	Role
View System	Used to build an application, including lists, grids, text boxes, buttons, and embedded web browser
Content Provider	Enabling applications to access data from other applications or to share their own data
Resource Manager	Providing access to non-code resources (localized strings, graphics, and layout files)
Notification Manager	Enabling all applications to display customer alerts in the status bar
Activity Manager	Managing the lifecycle of applications and providing a common navigation backstack

Android S/W Stack - Libraries

- Including a set of C/C++ libraries used by components of the Android system
- Exposed to developers through the Android application framework

Android S/W Stack - Runtime

- Core Libraries
  - ✓ Providing most of the functionality available in the core libraries of the Java language
  - ✓ APIs
    - Data Structures
    - Utilities
    - File Access
    - Network Access
    - Graphics
    - Etc

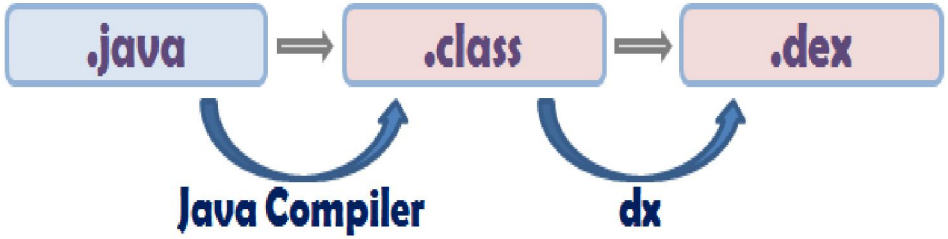
Android S/W Stack – Runtime (Cont)

- Dalvik Virtual Machine
  - ✓ Providing environment on which every Android application runs
    - Each Android application runs in its own process, with its own instance of the Dalvik VM.
    - Dalvik has been written such that a device can run multiple VMs efficiently.
  - ✓Register-based virtual machine

Android S/W Stack – Runtime (Cont)

Dalvik Virtual Machine (Cont)

- ✓Executing the Dalvik Executable (.dex) format
  - ➢.dex format is optimized for minimal memory footprint.
  - ➢Compilation



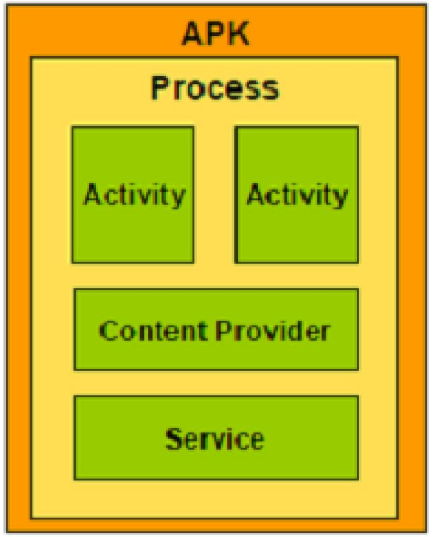
- ✓Relying on the Linux Kernel for:
  - ➢Threading
  - ➢Low-level memory management

Android S/W Stack – Linux Kernel

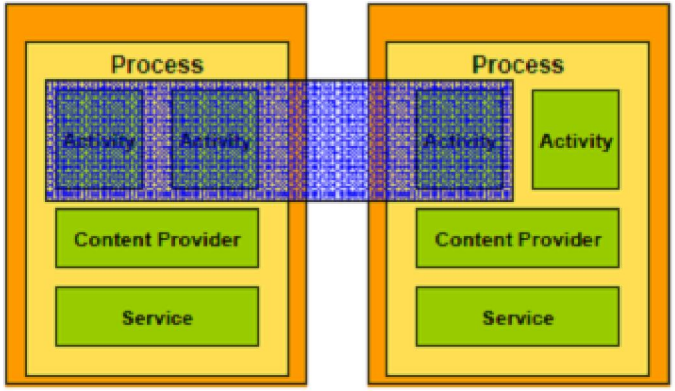
- Relying on Linux Kernel 2.6 for core system services
  - ✓Memory and Process Management
  - ✓Network Stack
  - ✓Driver Model
  - ✓Security
- Providing an abstraction layer between the H/W and the rest of the S/W stack

Android Component Model

- Packaging: APK File (Android Package)
  - Collection of components
  - Components share a set of resources
    - Preferences, Database, File space
  - Components share a Linux process
    - By default, one process per APK
  - APKs are isolated
    - Communication via Intents or AIDL
  - Every component has a managed lifecycle



- Task (what users know as applications)
  - Collection of related activities
  - Capable of spanning multiple processes
  - Associated with its own UI history stack
  - Processes are started & stopped as needed
  - Processes may be killed to reclaim resources



Android Application Components

- Activities – represent a single screen with a UI
- Services – represents a process running in the background
- Content Provider – a link back to the data
- Broadcast Receiver – listens for system-wide messages to respond to
- Application – a set of Activities that make up a cohesive unit
- Intent – a message to be passed

Activity

- Conceptually, an Activity is a single screen of your application
- In other words, an App really is a collection of related Activities
- Consider each Activity both a screen and a feature
- Apps can activate Activities in other Apps

Service

- A Service is a component that runs in the background to perform long-running operations
- A Service has no UI
- Examples of Services:
  - Playing music in background
  - Gathering GPS data
  - Downloading a data set from the server

Content Provider

- A Content Provider manages a shared set of app data
- This shared set of data could be a file, an SQLite DB, a remote link to a web service, etc.
- Apps can query a Content Provider for data if they have permission
- For example, your App could query the Contacts DB for a set of email addresses
- Content Providers can also be private

Broadcast Receiver

- A Broadcast Receiver responds to system-wide announcements (which are manifested as Intents)
- System status information is delivered this way (i.e. device turned on side, screen off, low battery, phone call incoming, etc.)
- Broadcast Receivers typically don't have a UI, but could have a status bar icon

Intent

- An Intent is a message that requests an action from another component of the system
- This includes the “please start up your App” Intent that the system sends when a user clicks on your App icon

Connected Apps

- Due to the component nature of Apps (made up of Activities, Services, etc.), it is easy to build features of your App using existing system components
- For example, if your App needs to take a picture, you can query the Camera Activity to handle that request and return the resulting image
- This is handled through Intents

Tying it all Together

- If an App is made up of all these disparate parts, what holds them all together?
- The AndroidManifest.xml file!
  - Sets up all permissions the user has to agree to (i.e. Internet, GPS, contacts, etc.)
  - Declares the API level of the App
  - Requests hardware features needed
  - Needed libraries
  - Which Activities are part of this App

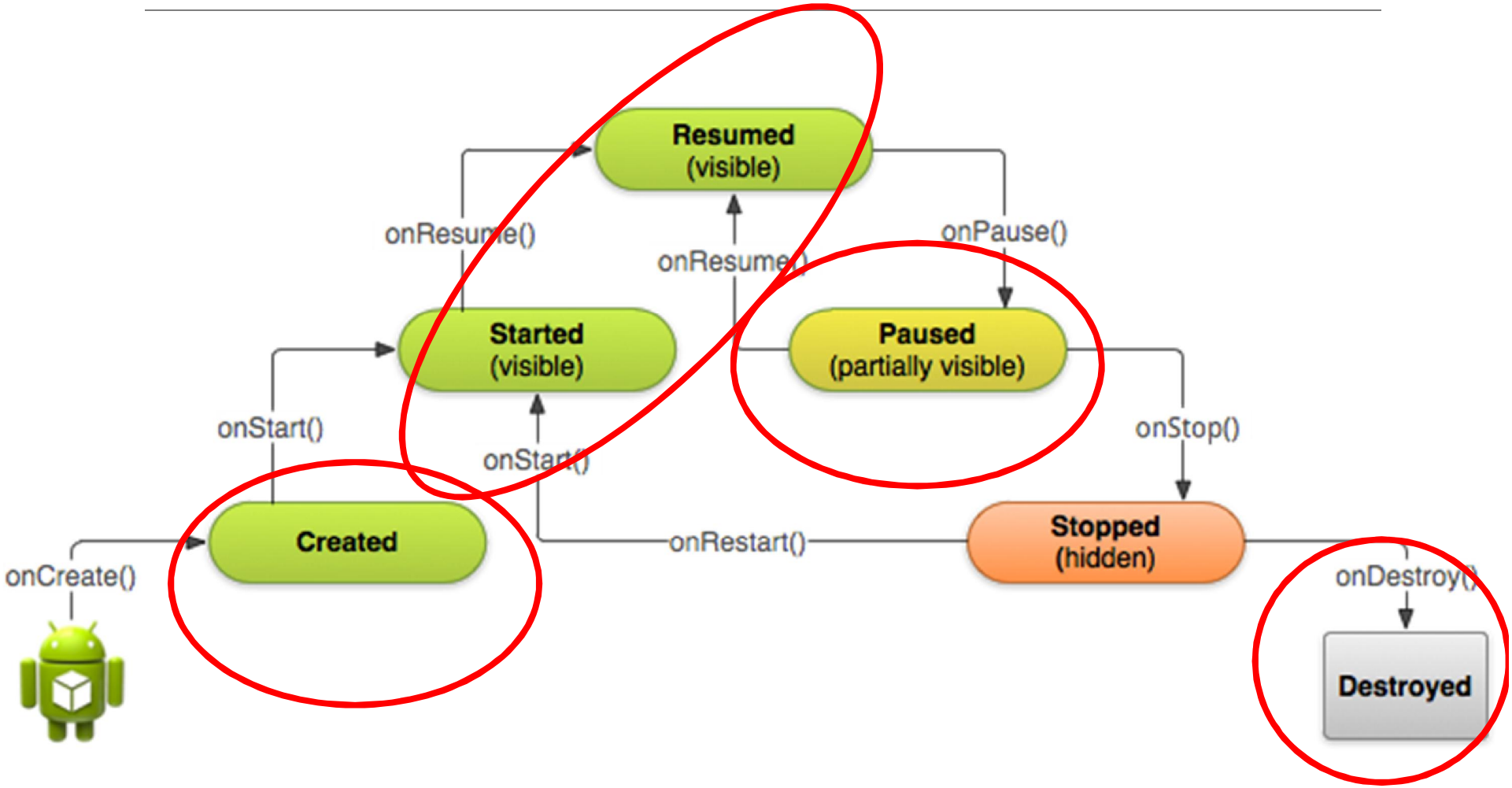
What about the other stuff?

- Typically referred to as “assets,” anything that isn’t code is placed in the res/ folder
- Music
- Images
- Some static data files

Where's the UI?

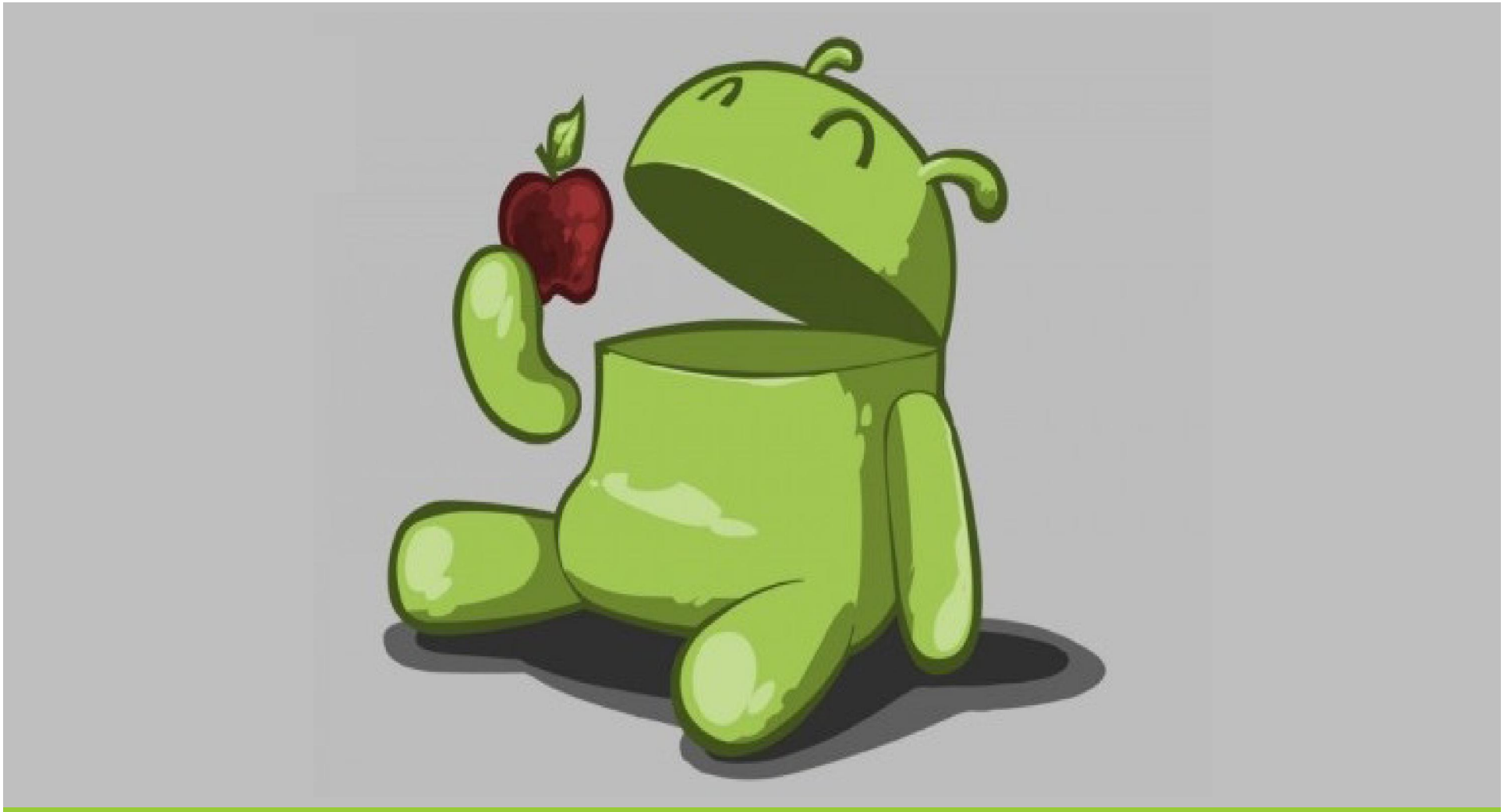
- The User Interface for an Android App is defined in the layout xml files
- Each layout xml file should correspond to an Activity

The App Lifecycle



Developing for Android

- Eclipse (NDA (Native Development Kit))
  - JDK (Java Development Kit) +JRE (Java Runtime Environment)
- Android SDK (Software Development Kit)
- Android Development Tools (ADT)
- Android Virtual Devices (AVD) & SDK Manager
- The Android Emulator
- Dalvik Debug Monitor Services (DDMS)
- The Android Debug Bridge (ADB)



Questions?