Week 7 – Mobile Application Programming

Mobile Devices

Mobile Devices

iPhone
- iPod touch (1,2,3,4)
- iPhone (2G, 3G, 3GS, 4, 4S, 5, 5C, 5S, 6, 6 Plus)
- iPad, iPad 2

Android
- Samsung (Galaxy)
- LG (Nexus, Optimus)
- HTC (One, OnePlus)
- Sony (Xperia)
- Motorola (Moto)
- Xiaomi (Mi, Redmi)

Windows Phone
- Nokia (Lumia)
- Acer (Liquid)

Blackberry

Mobile Devices

PCWorld | Windows phone | iOS | android

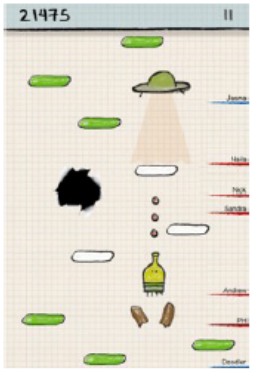
	Windows Phone 7	iOS (iPhone)	Android
Developer	Microsoft	Apple	Google
Copy/Paste	✘	✔	✔
Multitasking	✘	✔	✔
Flash Support	✘	✘	✔
Silverlight Support	✘	✘	✘
HTML5 Support	✘	✔	✔
Unified Inbox	✘	✔	✔
Exchange Support	✔	✔	✔
Threaded Email	✘	✔	✔
Visual Voicemail	✘	✔	✔
Video Calling	✘	✔	✔ Third Party App
Universal Search	✘	✔	✔
Internet Tethering	✘	✔	✔
Removable Storage	✘	✘	✔
Facebook Integration	✔	✘ (Third Party App)	✔ (Third Party Integration)
Twitter Integration	✘	✘ (Third Party App)	✔ (Third Party Integration)
Folders	Hubs	✔	✔
Apps Organization	Alphabetical	Customizable	Customizable
App Store	1,000+ Apps	300,000+ Apps	90,000+ Apps
Microsoft Office Support	Built-In	Third Party App	Third Party App
Widgets	Tiles on Home Screen	✘	✔
Media Sync	Zune Software Mac & PC	iTunes Mac & PC	Direct File Transfer + Third Party Software
X-Box Live Integration	Built-In	Via Third Party App	Via Third Party App

Mobile Devices

What's inside

- Multi-touch Screen
- Audio (speaker, microphone, vibrator)
- Connectivity (3G, Wifi, Bluetooth)
- Camera (front, rear, LED flash)
- Location (accelerometer, gyroscope, digital compass, assisted GPS)
- Sensor (volume, proximity, dock connector, etc.)

Types of Mobile Apps



Android Introduction

An operating system based on Linux kernel

A software platform for mobile devices

Allows writing application code in the Java language on Dalvik virtual machine

Open source under Apache Software License v.2

- No other licensing cost associated with software

Need to register on Google Play for publishing on device

- Android developer registration fee is US$25

Android Introduction

Developed by Google and later (a consortium of companies) the Open Handset Alliance (OHA)

- 14 mobile operators
- 23 handset manufacturers
- 21 semiconductor companies
- 17 software companies
- 12 commercialization companies

Unveiling of the Android platform was announced on 5-Nov-2007 with the founding of OHA

Android Version History

Release Date	API level	Version	Kernel	Codename	Icon
23 Sep 2008	1	1.0		Beta	
9 Feb 2009	2	1.1		Petit Four	
27 Apr 2009	3	1.5	2.6.27	Cupcake	
15 Sep 2009	4	1.6	2.6.29	Donut	
26 Oct 2009	5 ~ 7	2.0 ~ 2.1	2.6.29	Éclair	
20 may 2010	8	2.2	2.6.32	Froyo	
6 Dec 2010	9 ~ 10	2.3	2.6.35	Gingerbread	
22 Feb 2011	11 ~ 13	3.0 ~ 3.2	2.6.36	Honeycomb	
18 Oct 2011	14 ~ 15	4.0	3.0.1	Ice-cream Sandwich	
9 Jul 2012 13 Nov 2012 24 Jul 2013	16 17 18	4.1 4.2 4.3	3.0.31 3.4.0 3.4.39	Jelly bean	
31 Oct 2013	19 ~ 20	4.4		KitKat	
12 Nov 2014	21 ~ 22	5.0 ~ 5.1		Lollipop	

Android System Architecture

Android System Architecture

Linux Kernel

- Users never see the Linux subsystem
- It provides
  - Hardware abstraction layer
  - Memory management
  - Process management
  - Networking

Android System Architecture

System Libraries

- Bionic C library
  - BSD-derived implementation of the standard C system library (libc)
  - Optimized for embedded Linux-based devices
- SQLite
  - Lightweight relational database engine
- LibWebCore
  - Modern web browser engine which powers both the Android browser and an embedded web view
- FreeType
  - Bitmap and vector font rendering

Android System Architecture

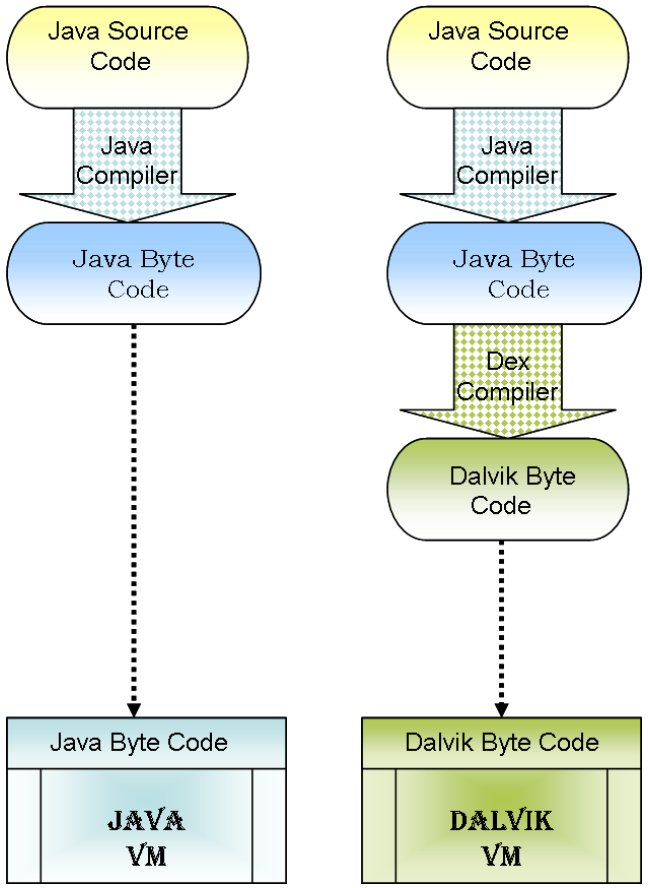
System Libraries

- Surface manager
  - Manage access to the display subsystem
  - Seamlessly composites 2D and 3D graphic layers from multiple applications
- SGL 2D graphics engine
- 3D libraries
  - An implementation based on OpenGL ES 1.0 APIs
  - Uses either hardware 3D acceleration or optimized software rasterizer
- Media libraries
  - Based on PacketVideo's OpenCore
  - Support playback and recording of audio/video formats and image files including MPEG4, H.264, AAC, AMR, JPG, PNG, etc.

Android System Architecture

Android Runtime

- Dalvik virtual machine
  - Google's implementation of Java
  - Optimized for mobile devices
- Key Dalvik differences
  - Register-based vs stack-based
  - Runs dex files
  - More efficient and compact implementation
  - Different set of Java libraries than SDK



Android System Architecture

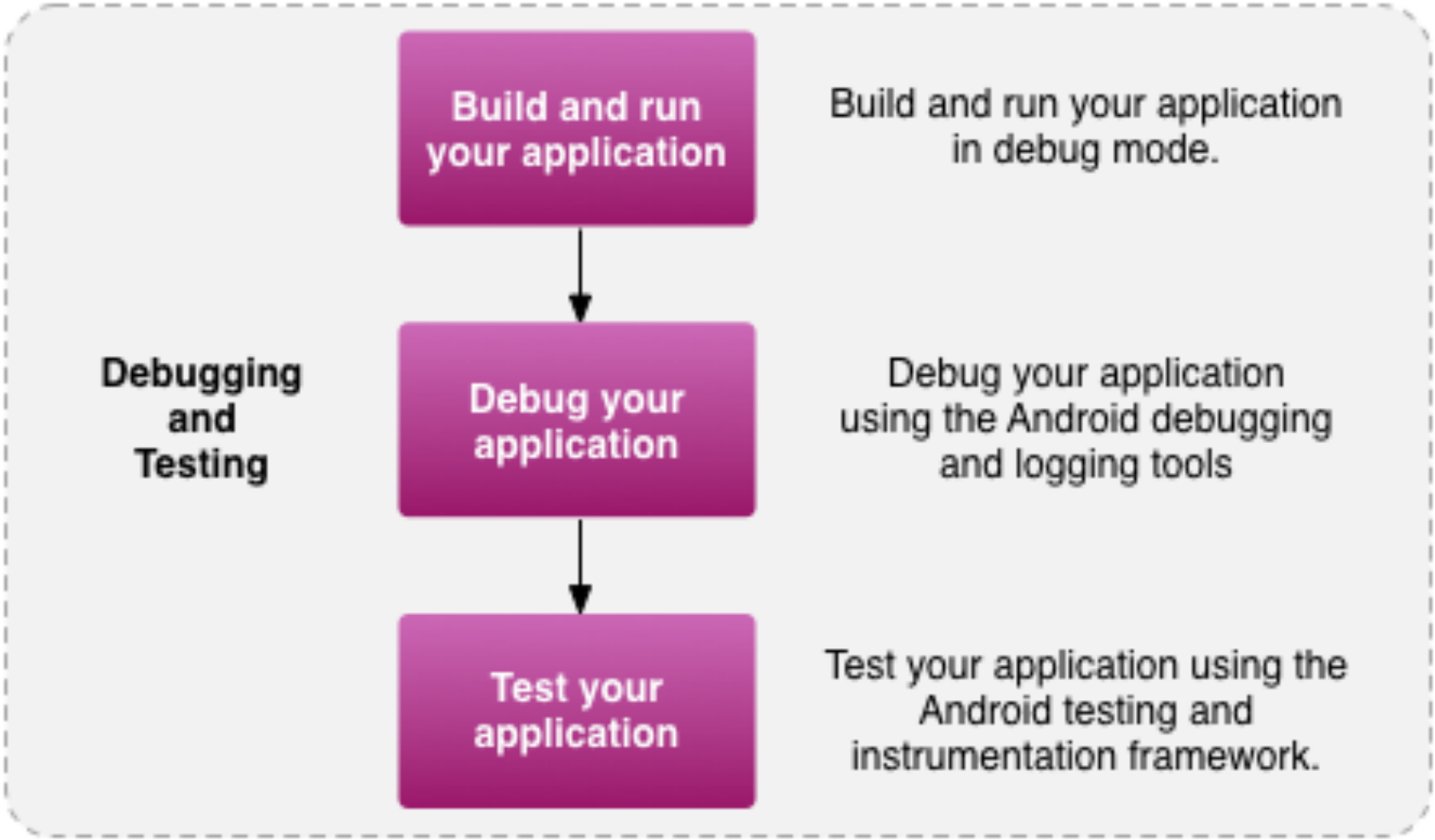
Application Framework

- Views
  - Basic GUI components including lists, grids, text boxes, buttons, and even embedded web browser
- Activity manager
  - Manages the life cycle of applications
  - Provides a common navigation feedback
- Content provider
  - Enables application to share their own data (e.g. Contacts) for other applications to access
- Resource manager
  - Provide access to non-code resources (e.g. strings, graphics, layouts, etc.)
- Notification manager
  - Enables all applications to display custom alerts in the status bar
- Location manager
  - Figures out the location of device through GSM, GPS, Wifi etc.

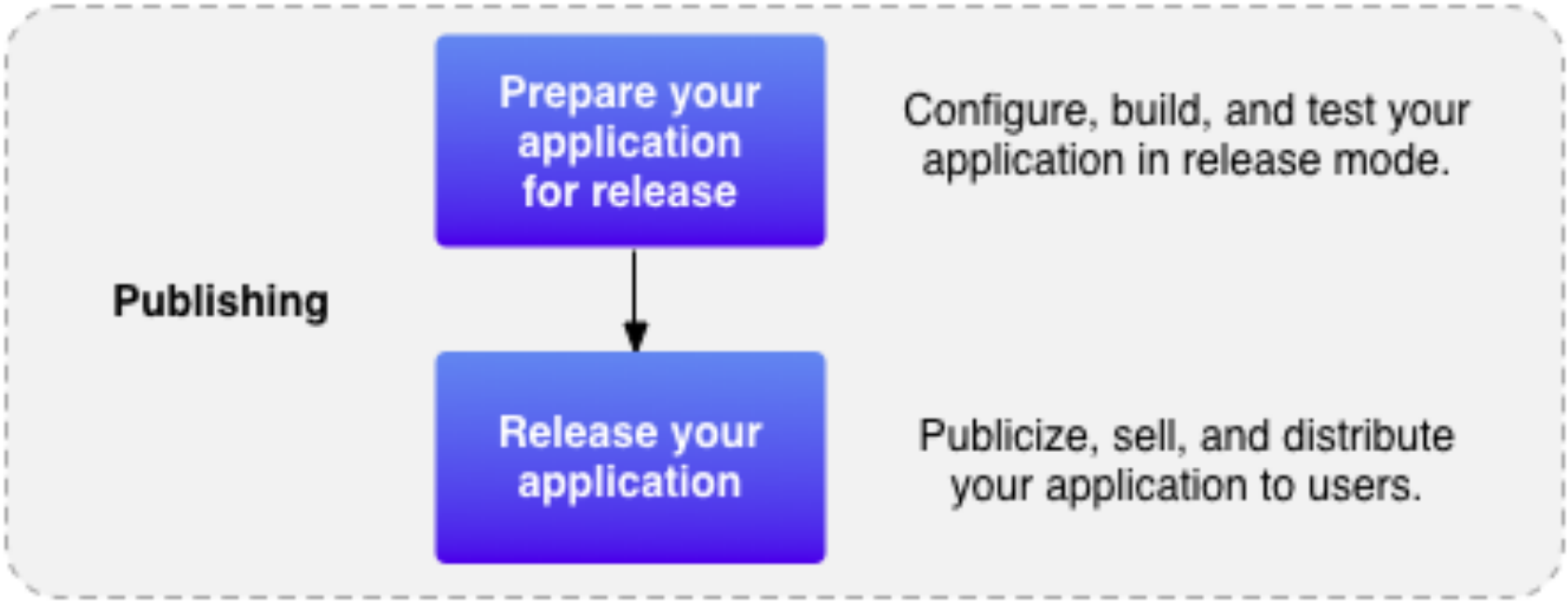
Android Developer Workflow

Android Developer Workflow

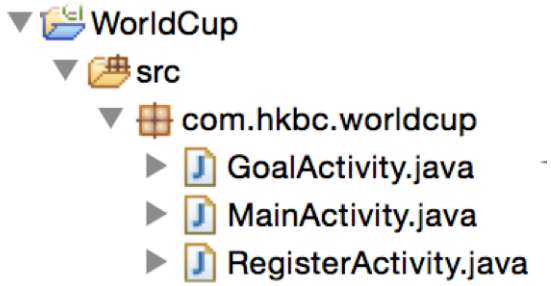
Android Developer Workflow



Android Developer Workflow

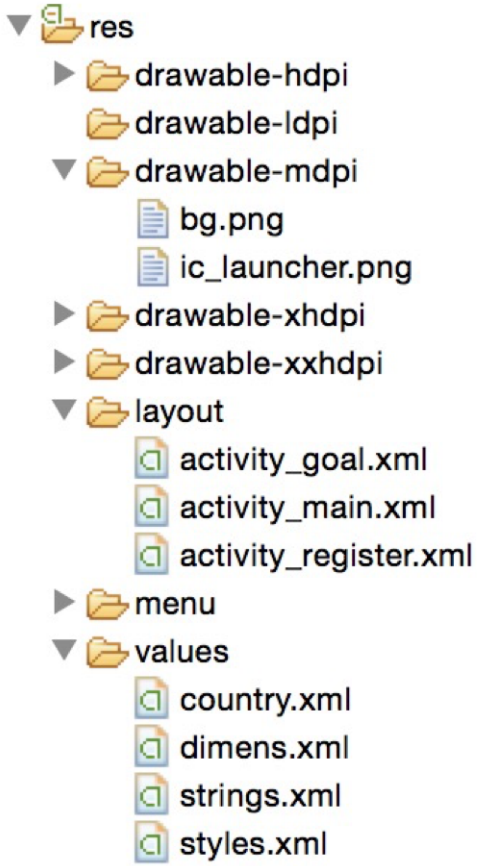


Android Project Structure



Source Code

- src/package_name/source_code.java
- event handling
- program logic



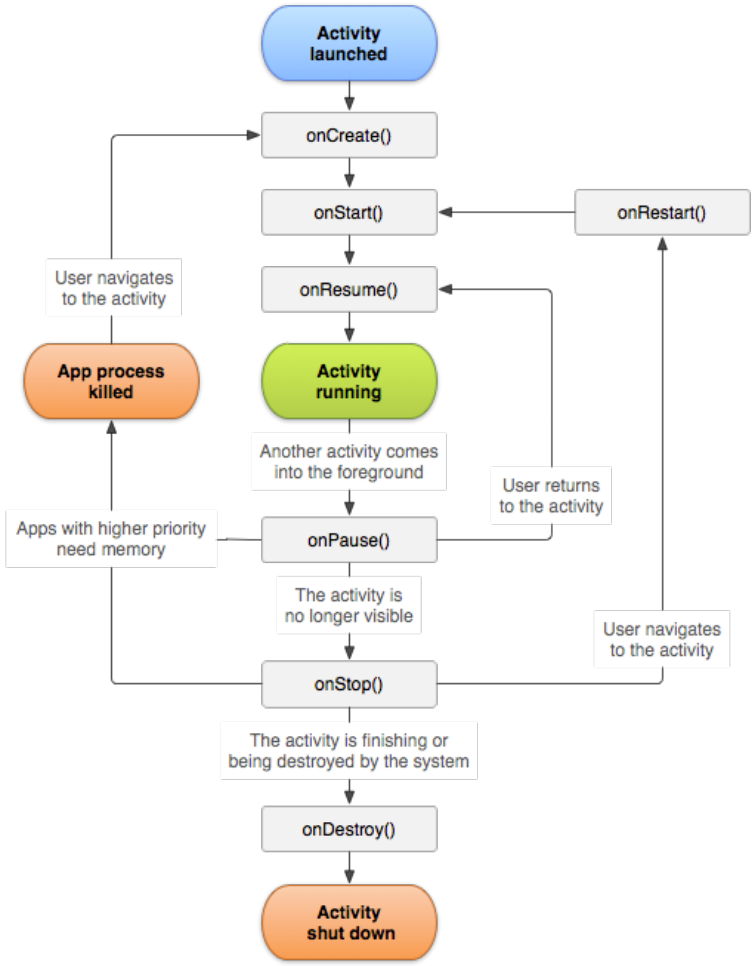
Resources

- res/drawable
  - Bitmap files (.png, .jpg, .gif)
- res/layout
  - XML files that define a GUI layout
- res/values
  - XML files that contain simple values
    - Strings
    - Integers
    - Colors

Android Components

Activity

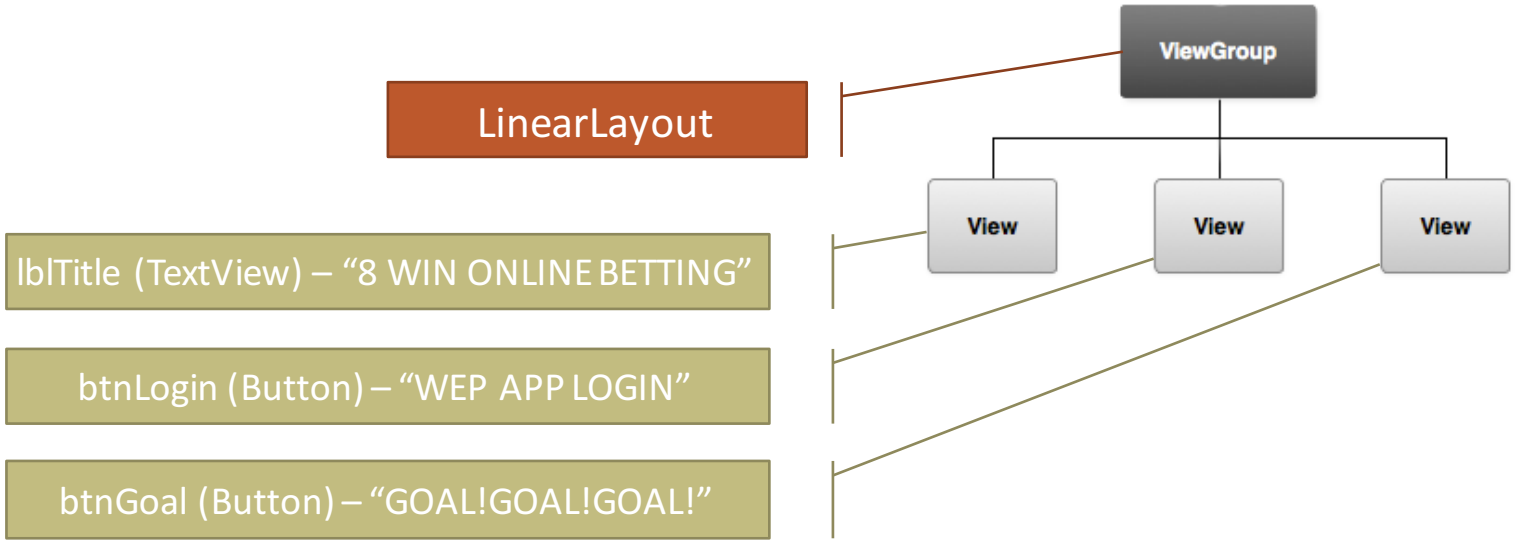
- A single screen with user interface
- Work together to form a cohesive user experience but are independent of each other
- An Android application can have several activities
- Android activity lifecycle



Android Components

View and View Group

- Also called as Layout Manager
- Extends the basic View class
- Can be nested to create a more complex layout
- View group is for arranging other views



Android Components

Intent

- Asynchronous message that activates other android components like activities, services and boradcast receivers
- Holds the content of the message
- Specify the name of the requested action and uniform resource identifier (URI) of the data to act on

Widgets

- Interactive compnents used on theAndroid (home screen)
- Display data and allow user to control

Android Components

Service (e.g. email application)

- Perform long-running operations in the background
- No user interface
- Continue to run in background even user switch to another application

Content Provider (e.g. user dictionary)

- Provide a structured interface to application data
- Share data between applications
- Used in conjunction with SQLite database for storing data to be accessed

Broadcast Receiver (e.g. low battery)

- Have to register the messages and intents first
- When specified event triggered, it will receive notification by Android system

Android Security and Permissions

Security Architecture

- A central design point is that no application, by default, has permission to perform any operations that would adversely impact other Apps, OS, or user
- An application's process is a secure sandbox and cannot disrupt other applications
- The permissions required by an application are declared statically in that application, so they can be know up-front at install time and will not change after that

Android Security and Permissions

Process Level Security

- Each Android application runs inside its own Linux process
- Each application has its own sandbox file system with its own set of preferences and database
- Other applications cannot access any of its data, unless it is explicitly shared
- Security enforcement happens at the process level, i.e. the code of any two packages cannot normally run in the same process, since they needed to run as different Linux users

Android Security and Permissions

File Level Security

- Each package (apk) file installed on the device is given its own unique Linux UID, creating a sandbox for it, and preventing it from touching other applications (or vice versa)
- This UID is assigned when the application is installed on the device, and remains constant for the duration of its life on that device
- Any data stored by an application will be assigned to that application's UID, and not normally accessible to the other packages
- The file created by a specific application is owned by it, but its global read/write permissions have been set appropriately so any other application can see it

Android Security and Permissions

Using Permissions

- A basic application has no permissions associated with it
- Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory, it presents
  - essential information about the app to the Android system
  - information that the system must have, before it can run any of the application's code
- To make use of protected features of the device, the AndroidManifest.xml or more <uses-permission> tags declaring the permissions must be included

References

Apple iOS developer

- https://developer.apple.com/resources/

Google Android developer

- http://developer.android.com/develop/