



CS 175

Mobile App Development

Introduction

Ramin Moazzeni
Aug 19, 2021

Agenda



Course Details

Mobile Computing Overview

Android Overview



Course Details

Course



Design and implementation of applications running on a mobile platform such as smart phones and tablets.

This CS 175 section uses Android.

Course Website and Syllabus



[SJSU Canvas](#)

Tentative Course Topics



AndroidStudio

SQLite

Security

Activities

Debugging

Layouts

Networking

App Market*

Views

Graphics*

Wearables*

Intents

Sensors

*NDK**

Background Srvs.

*RenderScript**

Lists

Location Services

*OpenGL**

Preferences

Tooling

* If time permits

Grading



Assignment	Component
Programming Assignments <ul style="list-style-type: none">• Assignment 1 (5%)• Assignment 2 (5%)• Assignment 3 (5%)• Assignment 4 (10%)• Assignment 5 (15%)	40%
Exercises	5%
Quizzes	5%
Exams (Midterm + Final)	50%

Important Dates



- Class Begins: Thursday, Aug 19
- Midterm: Tuesday, Oct 12
- Veteran's Day (**No Class**)
Thursday, Nov 11
- Thanksgiving Recess (**No Class**)
Thursday, Nov 25
- Final: Thursday, Dec 9

Academic Dishonesty



- Cheating and plagiarism will not be tolerated in this course.
- Any individual caught cheating on quizzes, homework, exercises, or the exams will be punished to the full extent under University rules including disciplinary action which may include dismissal from the University.
- Any work used without giving proper credit, it is considered plagiarism and cheating.
- At a minimum, any student caught cheating will receive no credit for the work concerned and will receive a reduction of one letter grade from their final course grade.
- Repeated/subsequent cheating and plagiarism stemmpts will result in a grade of **F** for the course.

Texts



- Online documentation: <http://developer.android.com/>
- Android Programming: The Big Nerd Ranch Guide 4th Edition, Bill Phillips, Brian Hardy
- The Busy Coder's Guide to Android Development (Mark L. Murphy)
<https://commonsware.com/Android/>



Mobile Computing Background

Mobile by the Numbers



7.4 billion people on the planet

6 billion people have a mobile phone

4.5 billion people have access to toilets

US Penetration



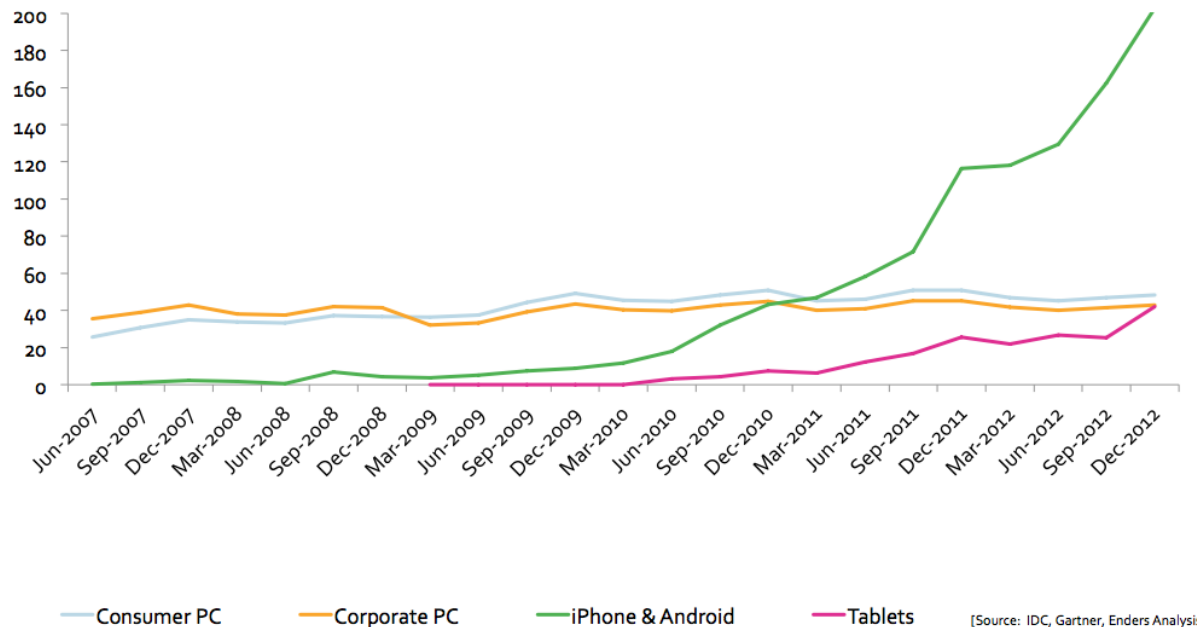
91% of all U.S. citizens have their mobile device
within reach 24/7



Mobile devices vs PC sales



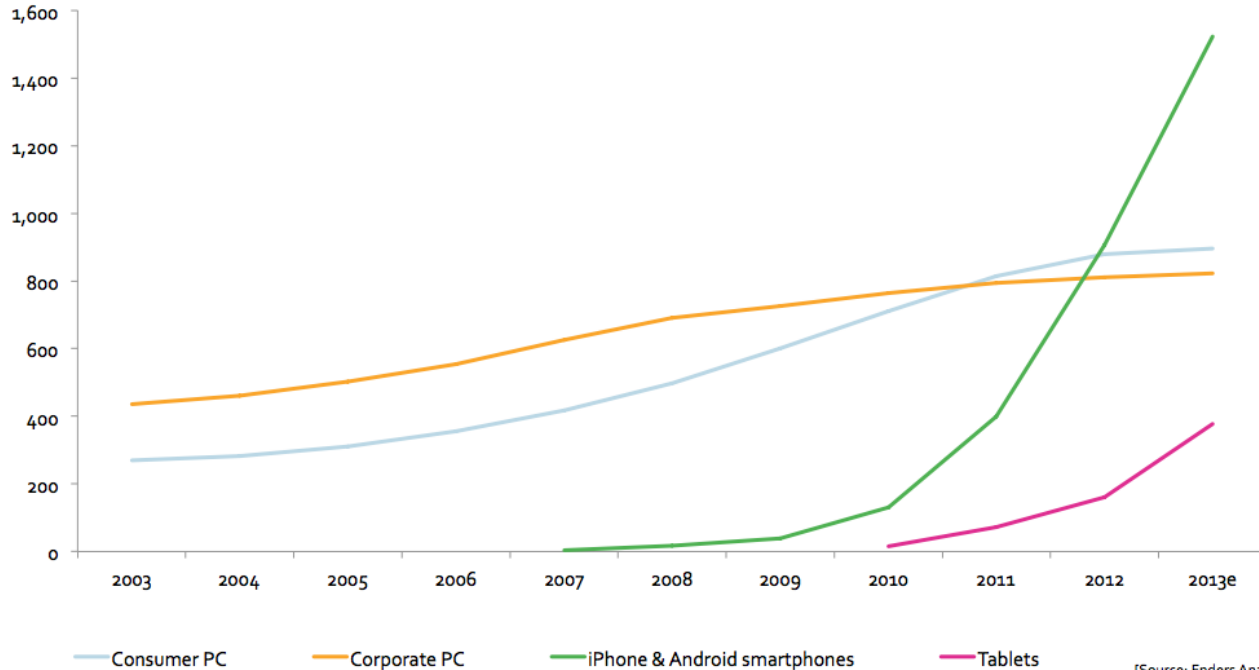
Quarterly unit sales (m)



Mobile devices vs PC install base



Global install base (m)



[Source: Enders Analysis]



Mobile Application Development

Few Reasons:

- Smart Phones
- Internet access anywhere
- Social networking
- Desire to share/record
- Millions of mobile users
- Open standards



Industry Effects

Consumer gadgets



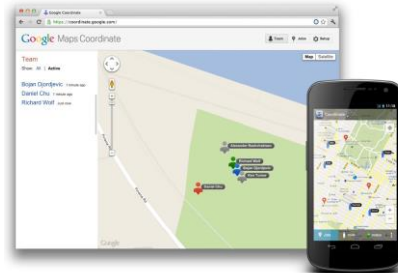
"Your smartphone is so smart it takes 14 gadgets and \$1,200 to match it"



Business gadgets



Kiosks
Coordination
Payments





Android Overview

What is the Android platform?



A free, open-source OS

An open-source development platform for creating apps

A complete stack – OS, Middleware, Apps

Powered by Linux OS

Fast application development in Java

Open source under the Apache 2 license

Devices that run Android

Key OS Features



Access to hardware (camera, GPS, sensors)

Data transfer (Bluetooth, NFC, Wi-Fi Direct)

Maps, geocoding, location-based services

Background services

SQLite

Shared data and Inter-Application communication

Key OS Features (cont'd)



Widgets and live wallpaper

Extensive media support and 2D/3D graphics

Cloud to Device Messaging (GCM)

Optimized memory and process management

... and much more ...

One app, different form factors



Android for different devices



ANDROID WEAR



PHONES



TABLETS



ANDROID TV



ANDROID AUTO

Android History



Started in 2003

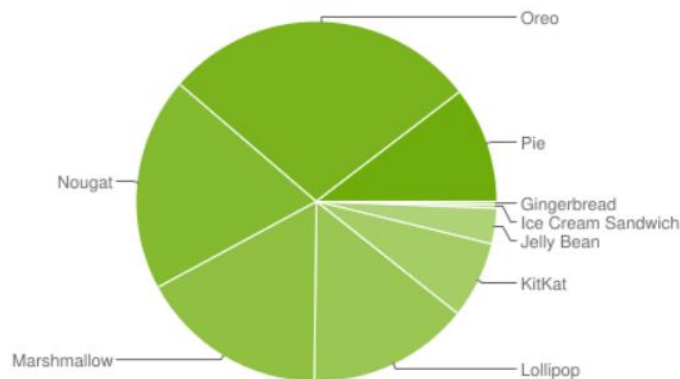
Acquired by Google in 2005

First Android phone (HTC Dream/G1) in 2008

First Android tablet (Motorola Xoom) in 2011

1 billion 30-day active users in 2014

Versions



Data collected during a 7-day period ending on May 7, 2019.
Any versions with less than 0.1% distribution are not shown.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%



Source:
<https://developer.android.com/about/dashboards/index.html>

Open Handset Alliance



Develops Android

84 technology and mobile companies

Why Android?



Powerful and open SDK

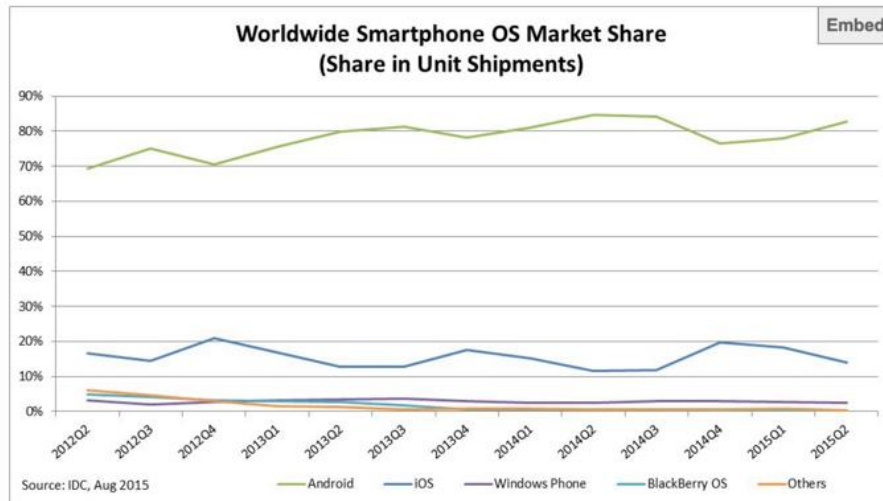
No licensing fees

Thriving developer community

Low barrier to entry

Huge potential market of users

Android Market Share



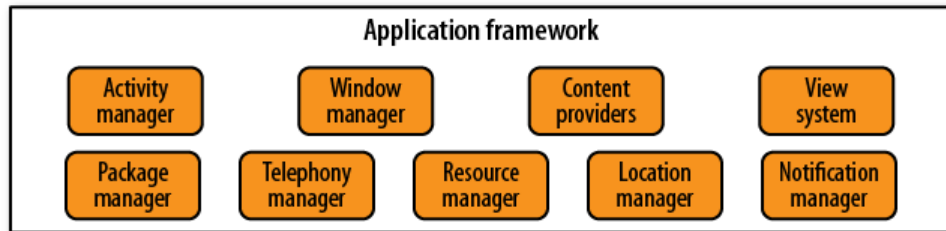
Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

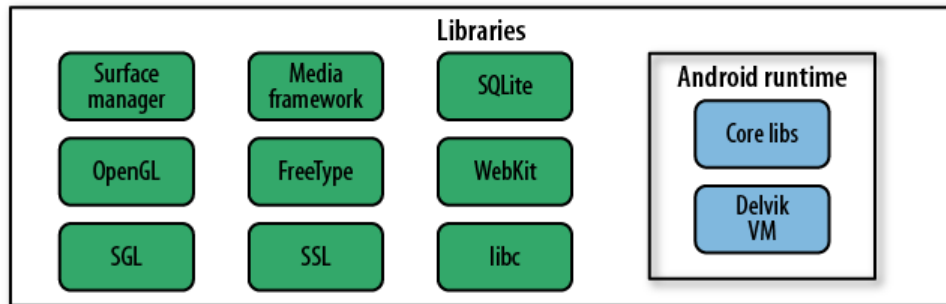
Stock apps, all of which can be replaced; modular architecture.



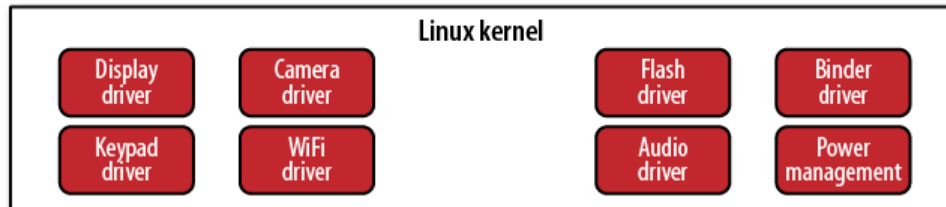
Managers (APIs to the OS). You will use these APIs in your apps.



SQLite and WebKit are in iOS as well. libc is from c. OpenGL is graphics, etc.



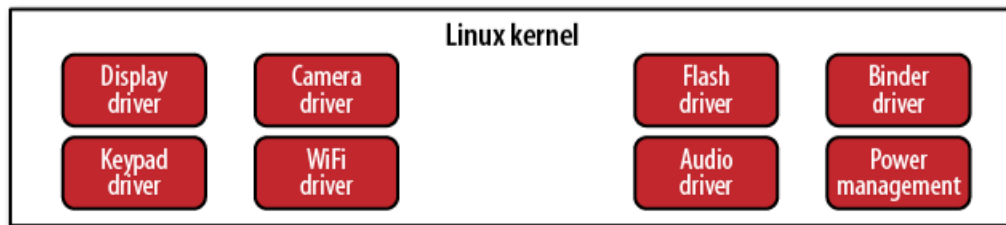
kernel components; you won't touch these.



Linux Kernel



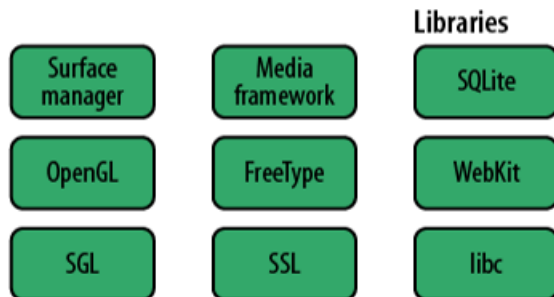
- Works as a HAL
- Device drivers
- Memory management
- Process management
- Networking



Libraries



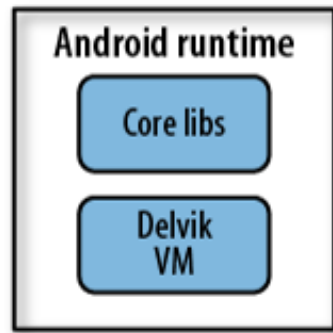
- C/C++ libraries
- Interface through Java
- Surface manager – Handling UI Windows
- 2D and 3D graphics
- Media codecs, SQLite, Browser engine



Android Runtime



- Dalvik VM
 - Dex files
 - Compact and efficient than class files
 - Limited memory and battery power
- Core Libraries
 - Java Std edition
 - Collections, I/O etc...



Dalvik



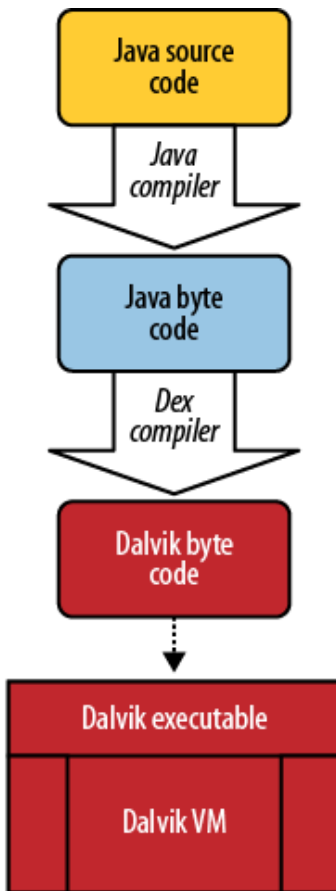
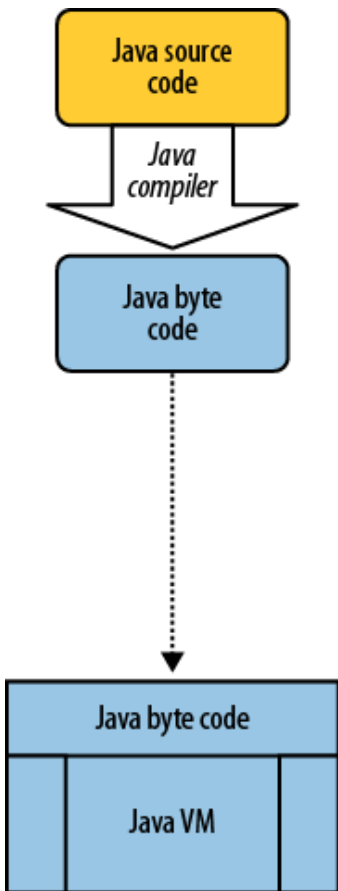
Virtual machine on Android

JIT compilation, registered-based

Apps: Java -> Java class file -> .dex file.

Designed for systems with CPU and memory constraints

Video: [Dalvik Virtual Machine Internals](#)

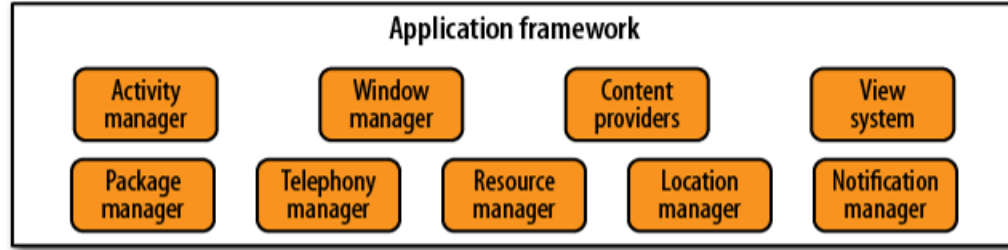


There is an extra step in the toolchain.

Reflection metadata is stripped out to make the class files smaller and more efficient.

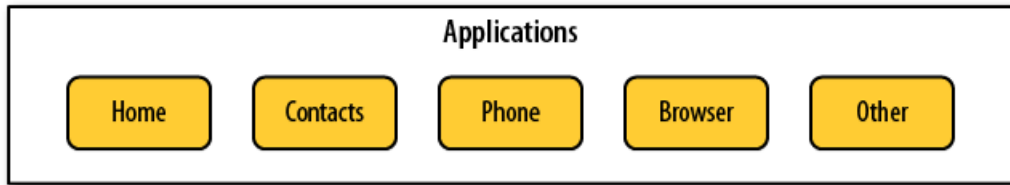
android dex files are like java class files - they're bytecode, but they've been optimized for a resource constrained environment.

Application Framework



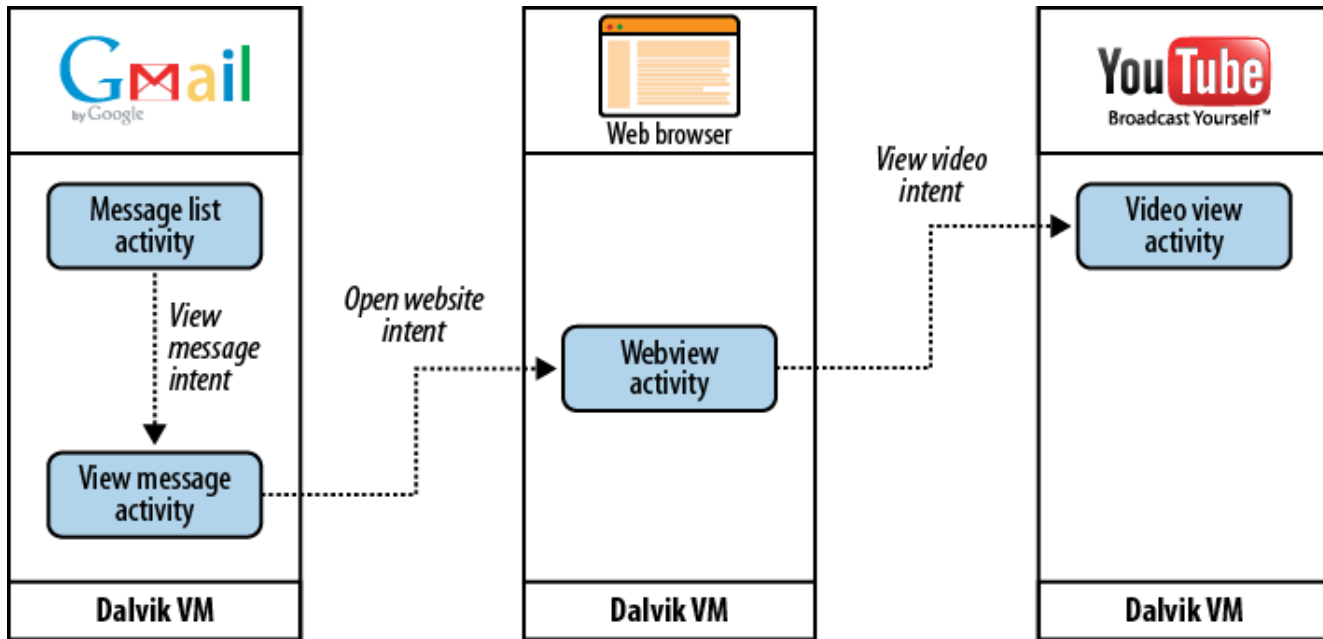
- API interface
- Activity manager – manages application life cycle.

Applications



- Built in and user apps
- Can replace built in apps

Android is modular



Android forces you to be modular.

Application Building Blocks



Activity

IntentReceiver

Service

ContentProvider

Activities



Typically correspond to one UI screen

But, they can:

- Be faceless

- Be in a floating window

- Return a value

IntentReceivers



Components that respond to broadcast 'Intents'

Way to respond to external notification or alarms

Apps can invent and broadcast their own Intent

Intents



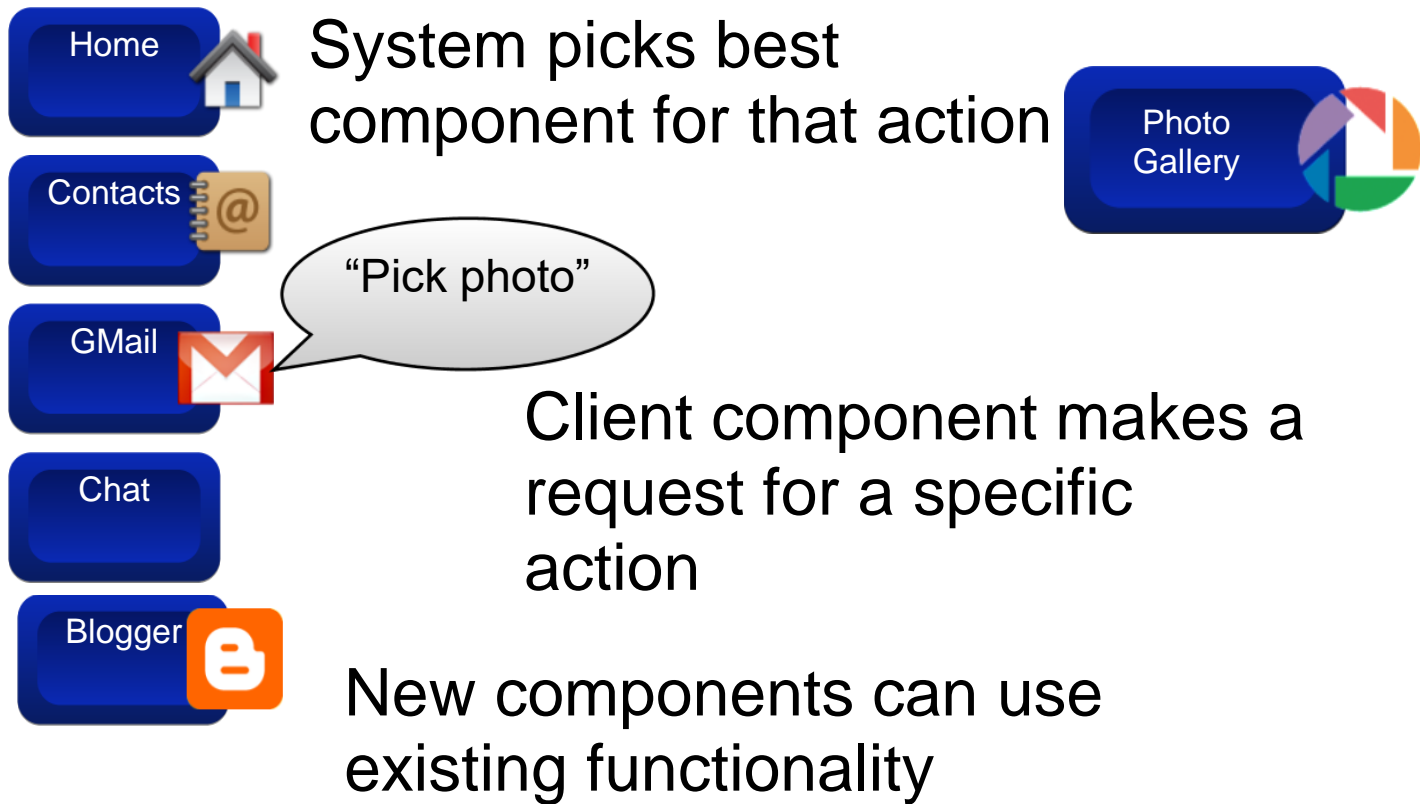
Think of Intents as a verb and object; a description of what you want done

E.g. VIEW, CALL, PLAY etc..

System matches Intent with Activity that can best provide the service

Activities and IntentReceivers describe what Intents they can service

Intents



Services



Faceless components that run in the background

E.g. music player, network download etc...

ContentProviders



Enables sharing of data across applications

E.g. address book, photo gallery

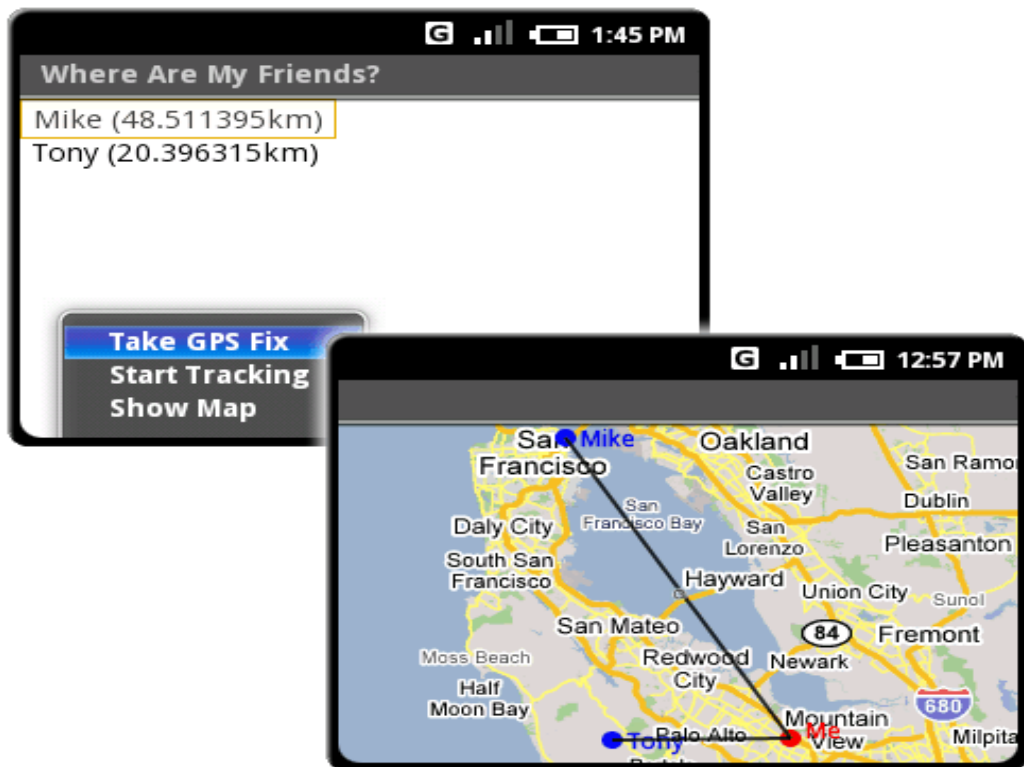
Provides uniform APIs for:

querying

delete, update and insert.

Content is represented by URI and MIME type

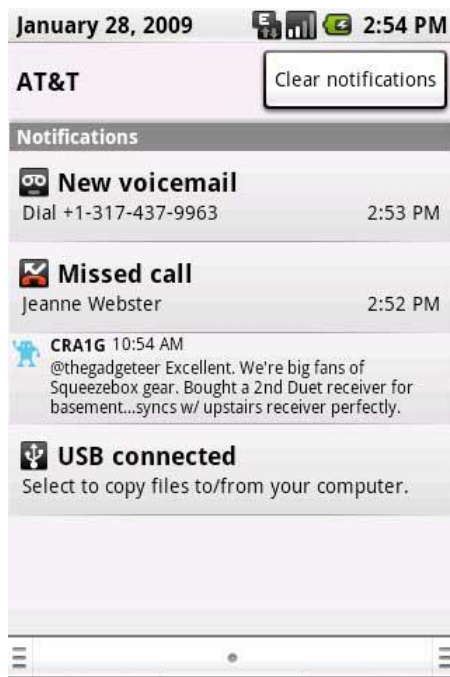
Location Manager



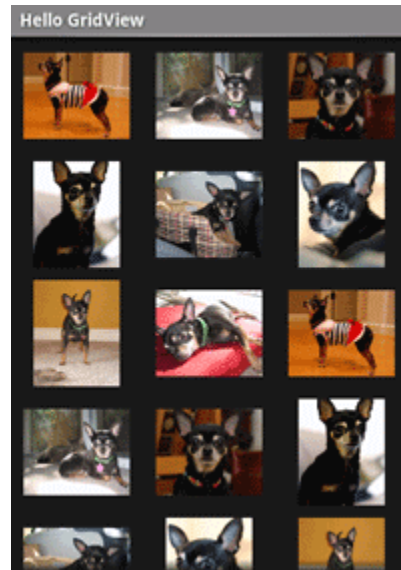
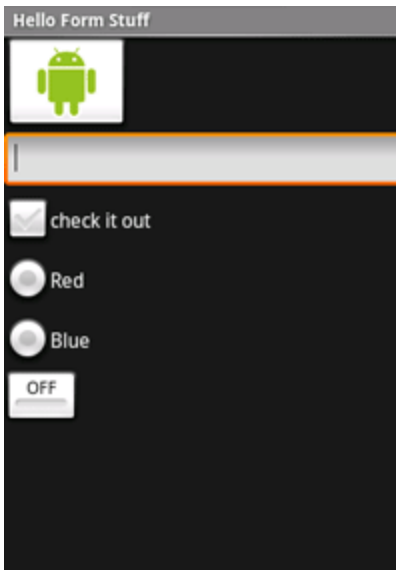
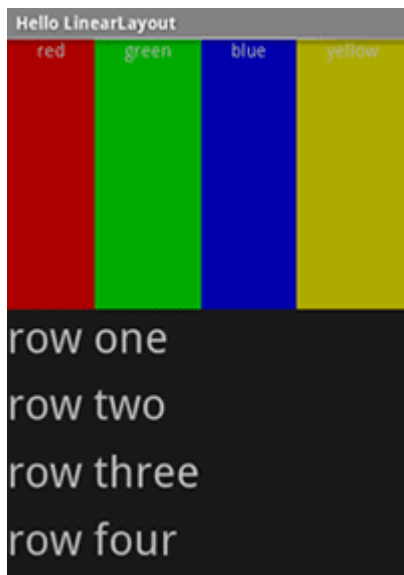
Notification Manager



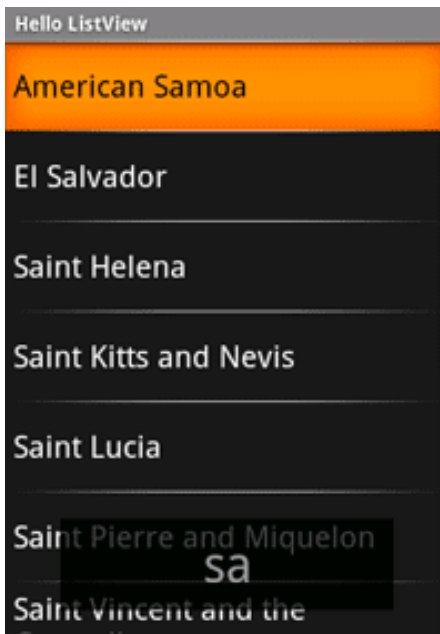
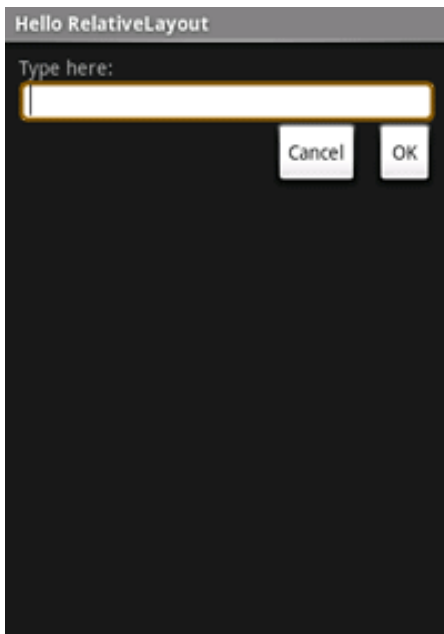
- How background app interact with users
- Consistent notification presentation



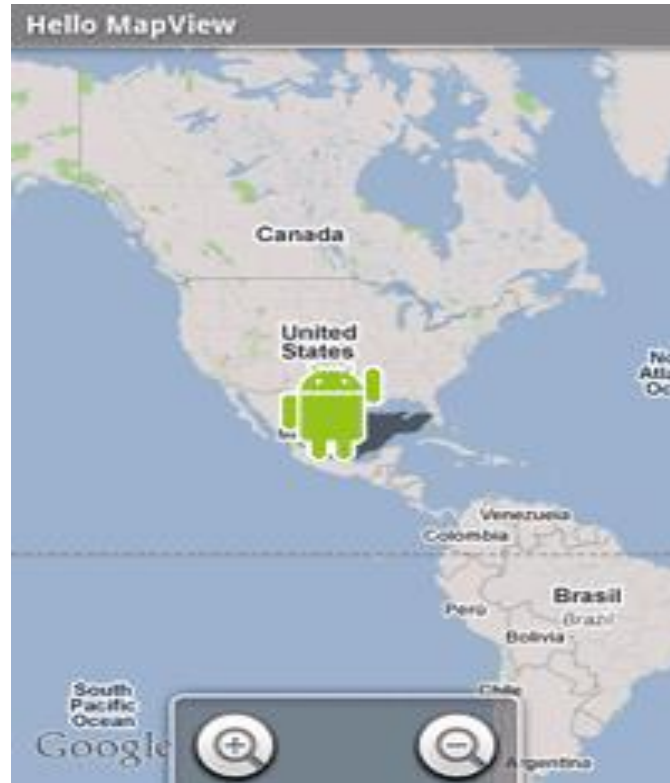
Views



Views



Location Manager



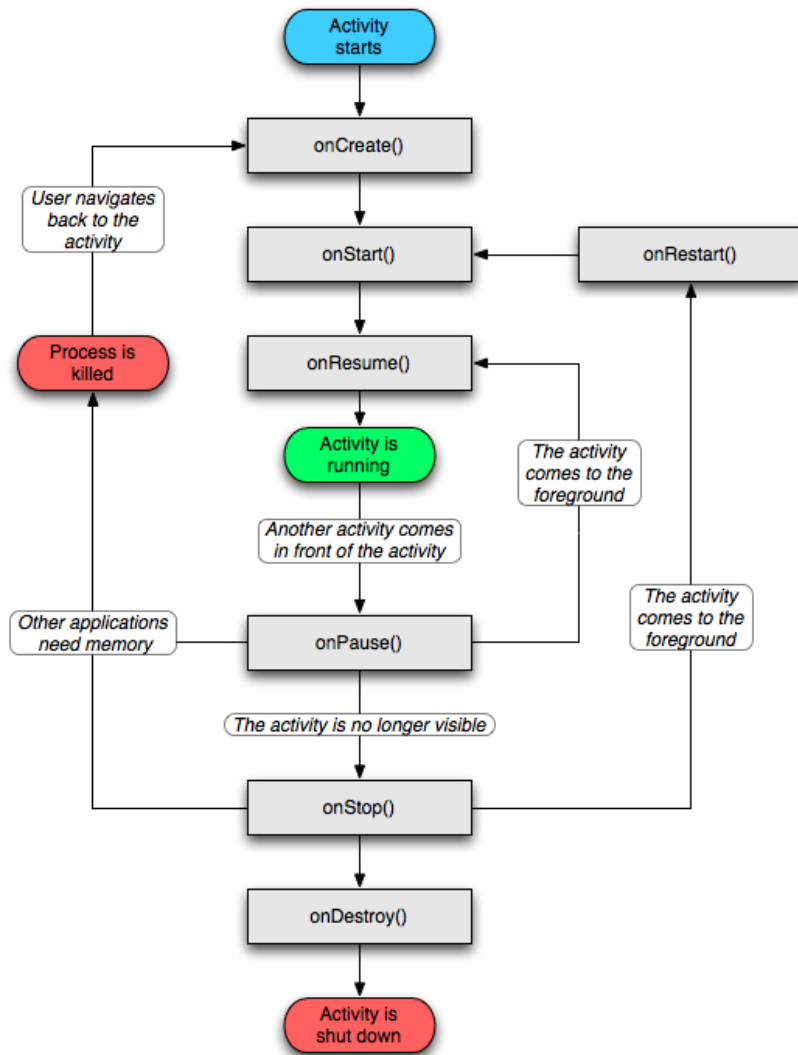
Application Lifecycle



Application run in their own processes (VM, PID)

Processes are started and stopped as needed to run an application's components

Processes may be killed to reclaim resources



Development Tools



Android SDK

- Provides the Java framework classes
- <https://developer.android.com/studio/releases/sdk-tools>

Android platform tools

- adb(android debug bridge): debug apps from your dev system

Android Developer Tools

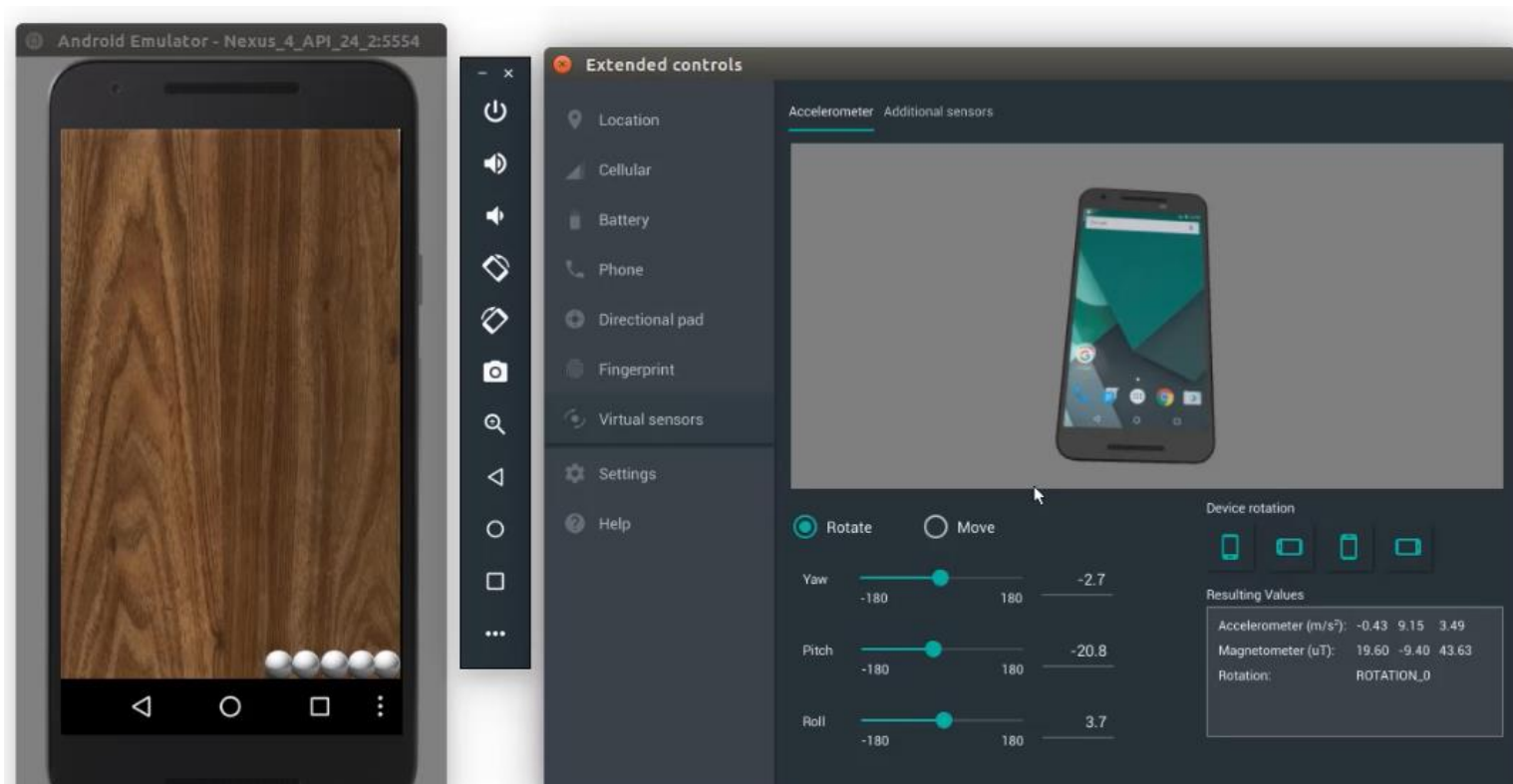
- Android Studio is the official IDE for apps development
- <https://developer.android.com/studio>



The Emulator

- The Emulator simulates Android devices on your computer
 - you can test your application on a variety of devices and Android API levels
 - Runs the same image as the device
 - No need to have each physical device.
- Limitations:
 - Bluetooth
 - NFC
 - Device-attached headphones
 - USB

The Emulator



Devices



Hello World



A great starting point:

<https://developer.android.com/training/basics/firstapp>

Generating UIs

Views – building blocks

E.g. TextView, EditText, Button

Placed into Layouts

E.g. LinearLayout, GridLayout, RelativeLayout