

Android 101 Workshop

First Android Workshop in Biratnagar, May 3rd 2014



Presented By

Bhupal Sapkota

CTO Semicolon Developers
Community Manager
GDG Kathmandu

Hem Shrestha

Android Developer
at PicoVico / SparrowSMS

Samrakchhan Ghimire

Sr. Android Developer
Co-founder
BirthdayForest

Table of Content

Android Platform Basics

Development Environment

Android App Basics

Android UI Design

Android Storage

Working with Web Services (JSON)

Perquisites

Basic Java

(If else, loops etc, Data Types, Data structures)

Object Oriented Programming

(Class, Object, Inheritance)

Android Platform Basics

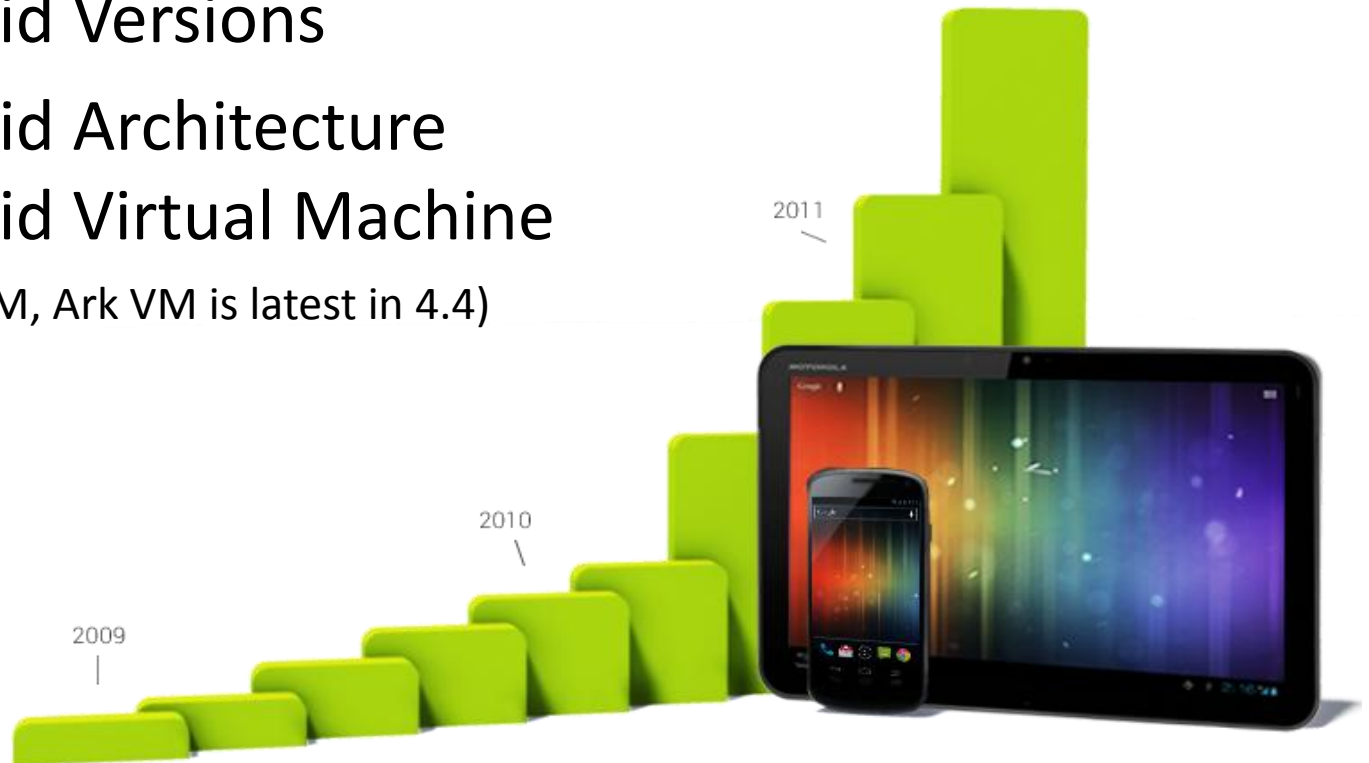
Introduction and history

Android Versions

Android Architecture

Android Virtual Machine

(Dalvik VM, Ark VM is latest in 4.4)



Introduction

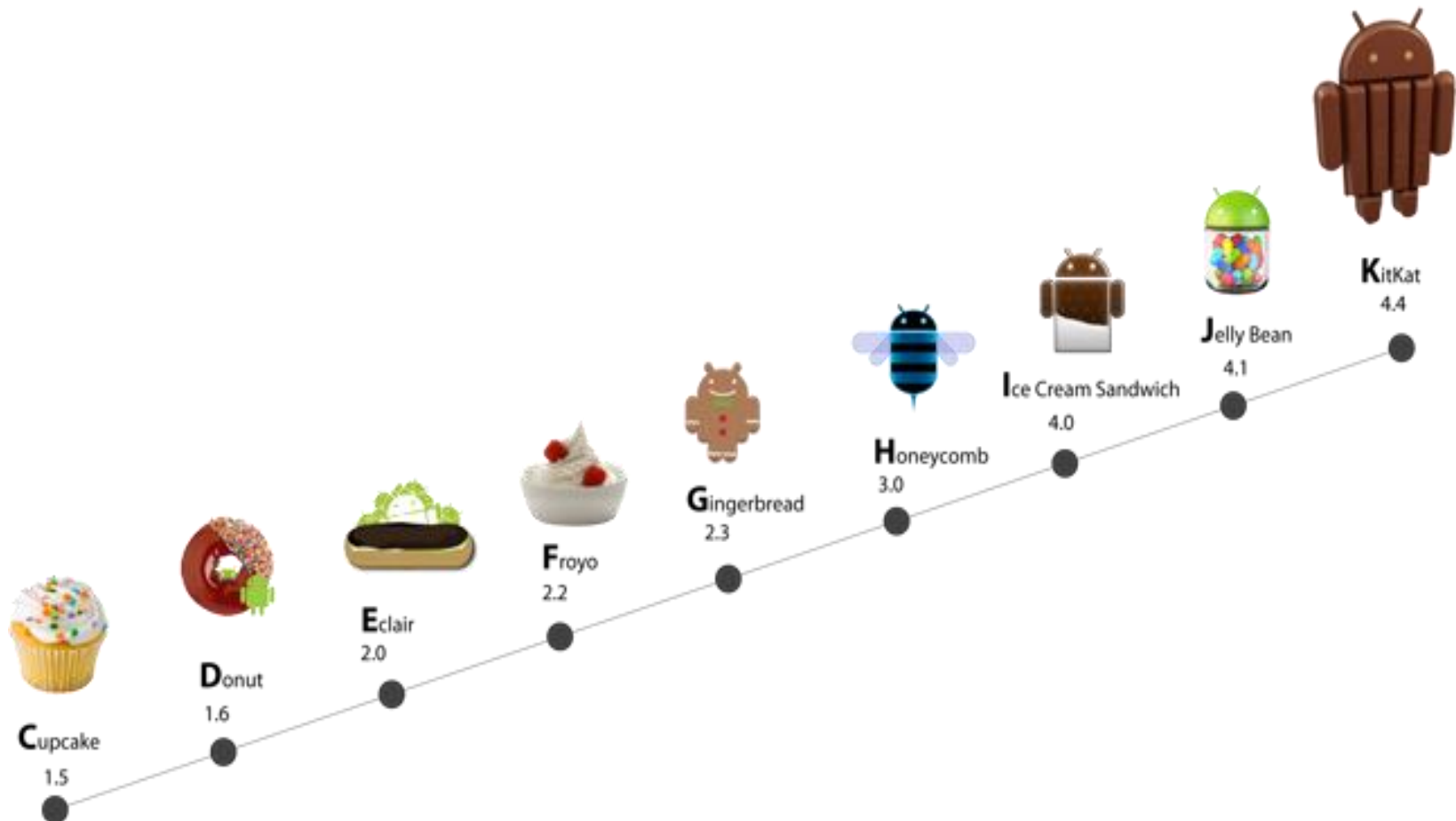
A Linux based Operating System designed primarily for touch screen mobile devices

Initially developed by Android Inc, lead by Andy Rubin and later purchased by Google in 2005

Android is Open Source and Google releases code under Apache2 license

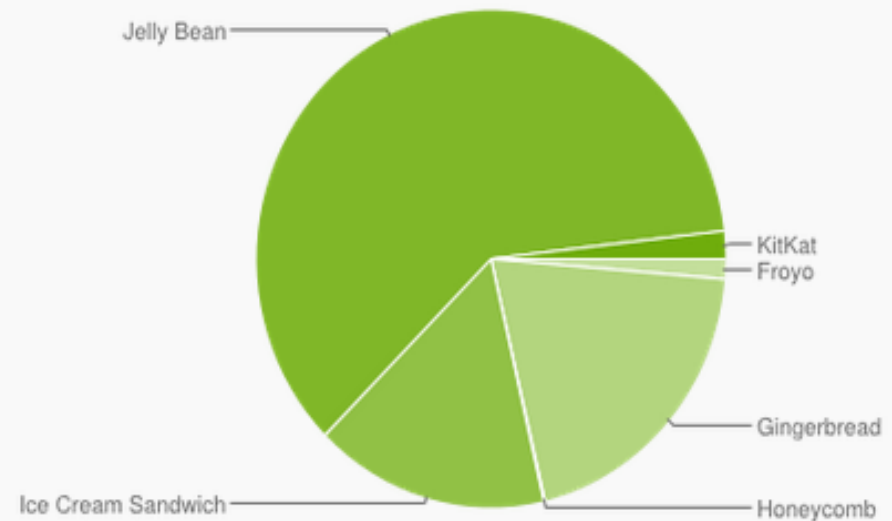


Android Versions



Android Versions Distribution

Version	Codename	API	Distribution
2.2	Froyo	8	1.3%
2.3.3 - 2.3.7	Gingerbread	10	20.0%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	16.1%
4.1.x	Jelly Bean	16	35.5%
4.2.x		17	16.3%
4.3		18	8.9%
4.4	KitKat	19	1.8%

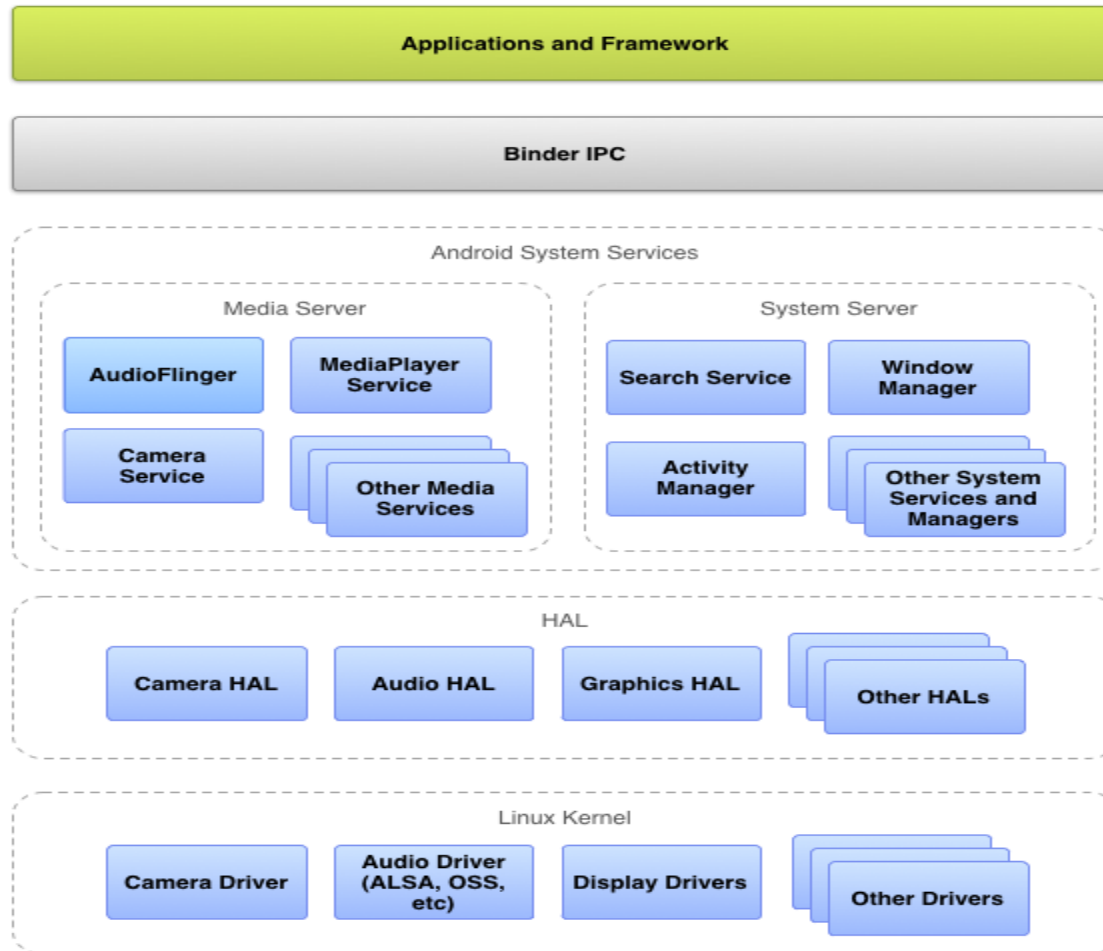


*Data collected during a 7-day period ending on February 4, 2014.
Any versions with less than 0.1% distribution are not shown.*

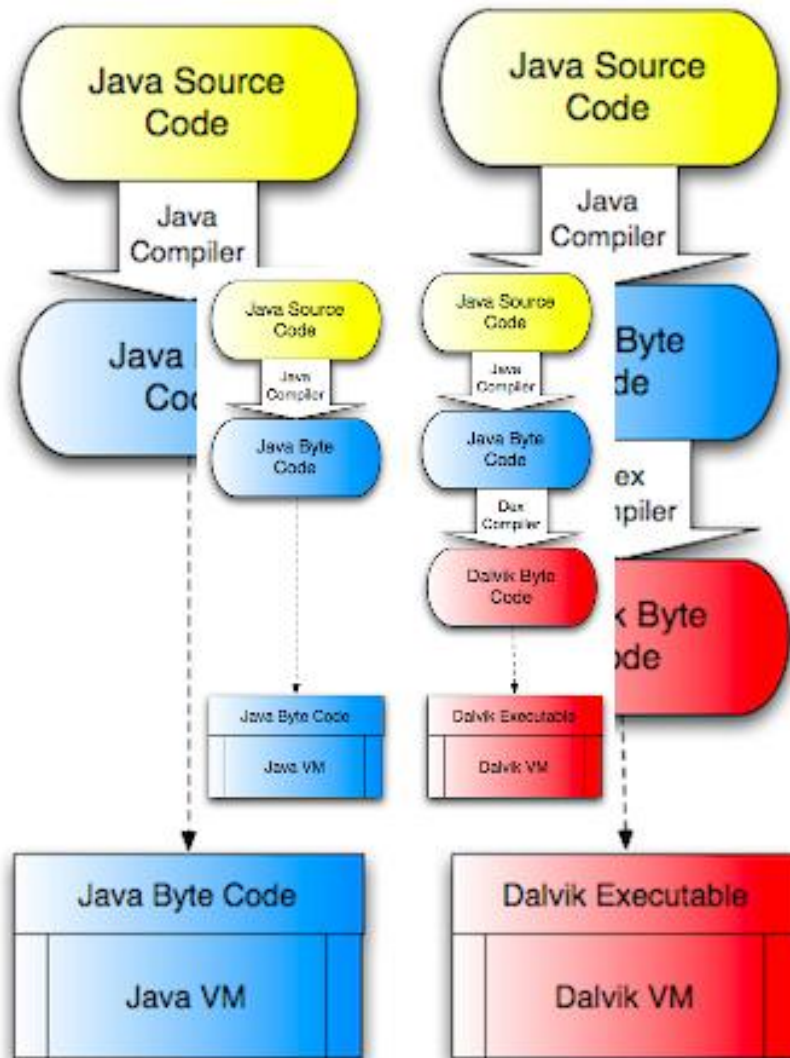
Android Architecture



Android Architecture



Dalvik Virtual Machine



Experimental
Art VM since
Android 4.4 /
KitKat

Android Development Environment



Android Development Environment

Required Software

JDK

Eclipse

Eclipse ADT Plugin

Android SDK

or Android SDK Bundle from Google

(Includes Eclipse, ADT, Android SDK)

Android SDK Manager

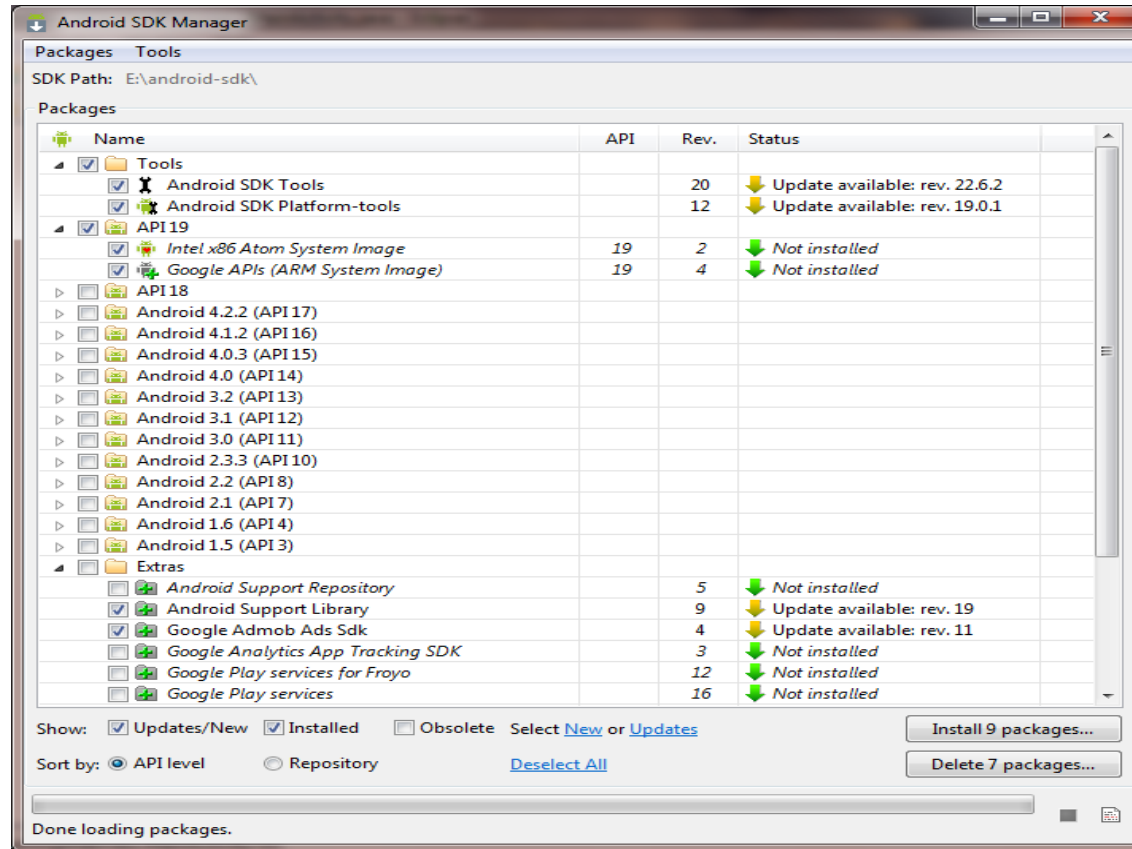
Android Virtual Device (AVD)

Debugging with Android LogCat

Eclipse Perspective – Java/DDMS



Android SDK Manager

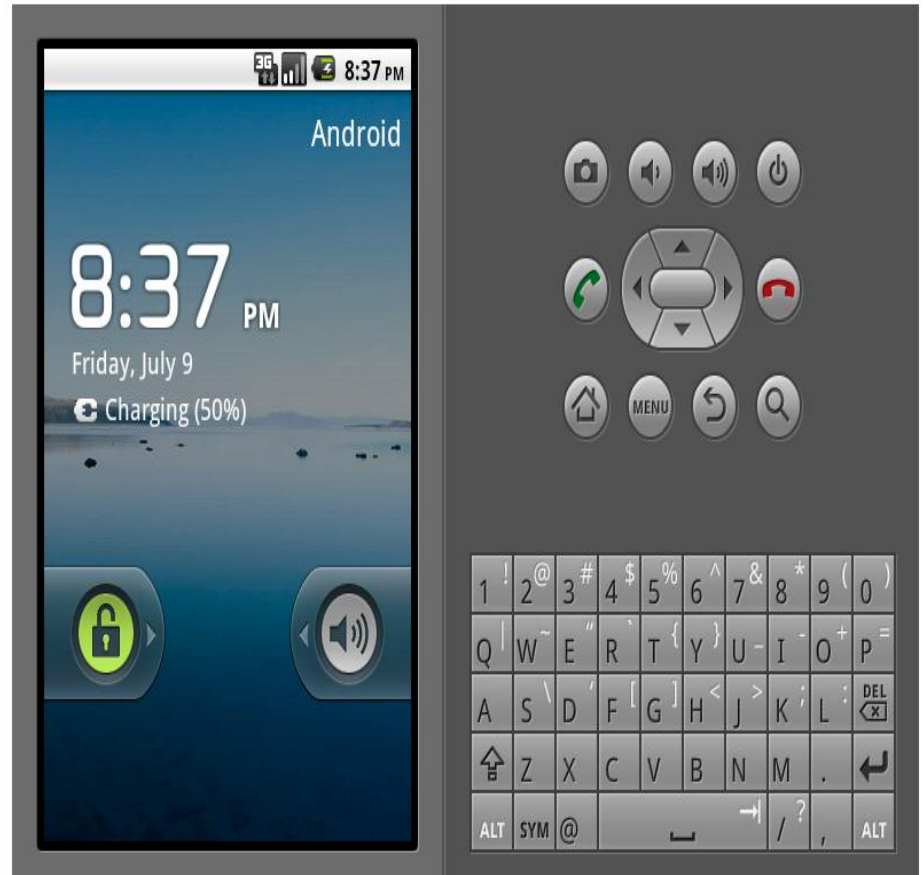


Eclipse (with ADT), Window Menu > Android SDK Manager

Android Virtual Device(AVD)



Real Device



Virtual Device

Android Virtual Device(AVD)

Create new Android Virtual Device (AVD)

Name:

Target:

CPU/ABI:

SD Card:

☒ Size: MiB

☐ File:

Snapshot:

☐ Enabled

Skin:

☒ Built-in:

☐ Resolution: x

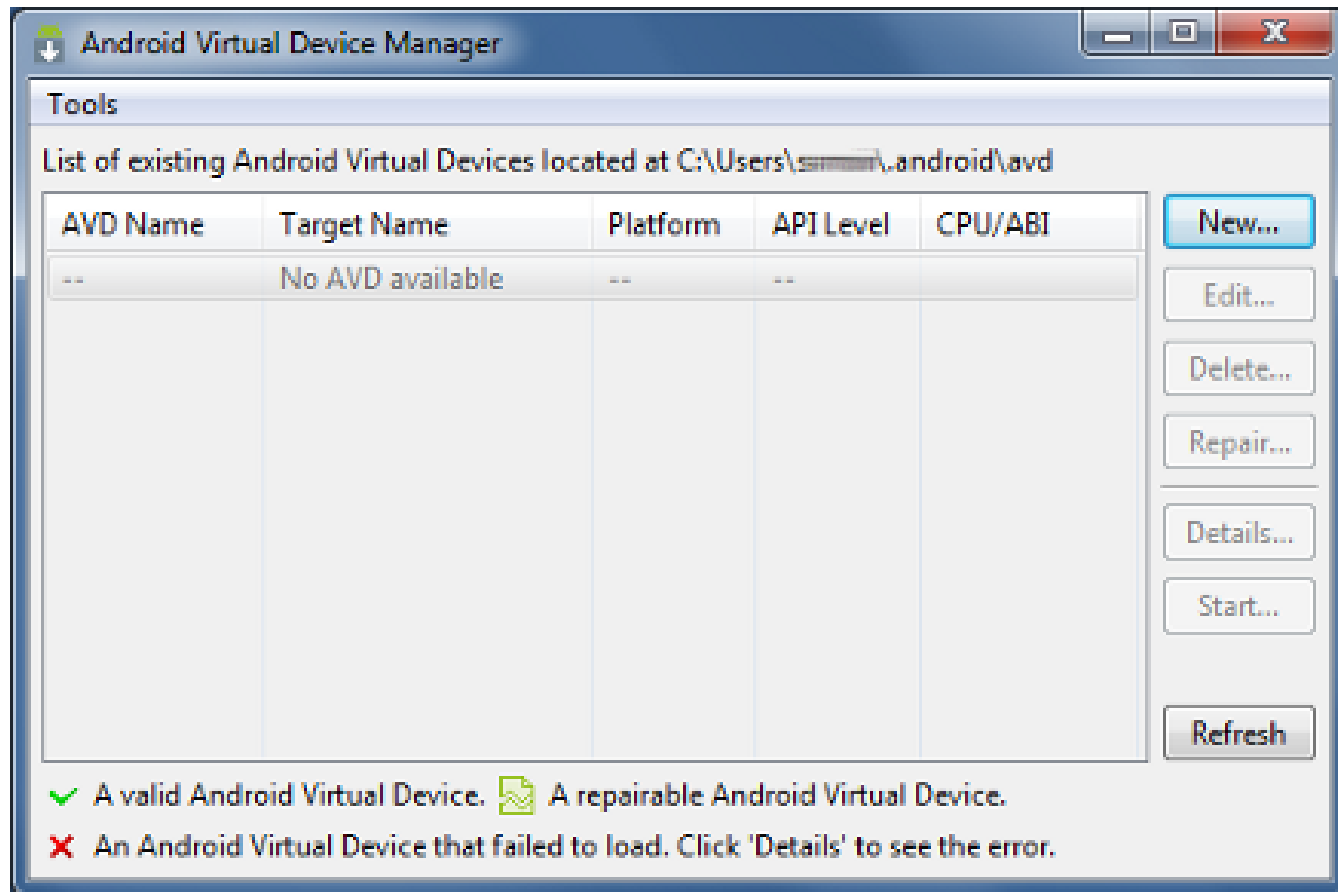
Hardware:

Property	Value	

☐ Override the existing AVD with the same name

AVD Create Dialog

Android Virtual Device(AVD) Manager



In Eclipse: Window Menu > AVD Manager

Android LogCat

The screenshot displays the Android Studio IDE. The top editor shows the `MainActivity.java` file with the following code:

```
package com.at.androidtraining1;

import android.os.Bundle;

public class MainActivity extends Activity implements OnClickListener{

    private Button btnListview, btnStyling, btnDatabase, btnSharedPreferences, btnWebview, btnWebServices;

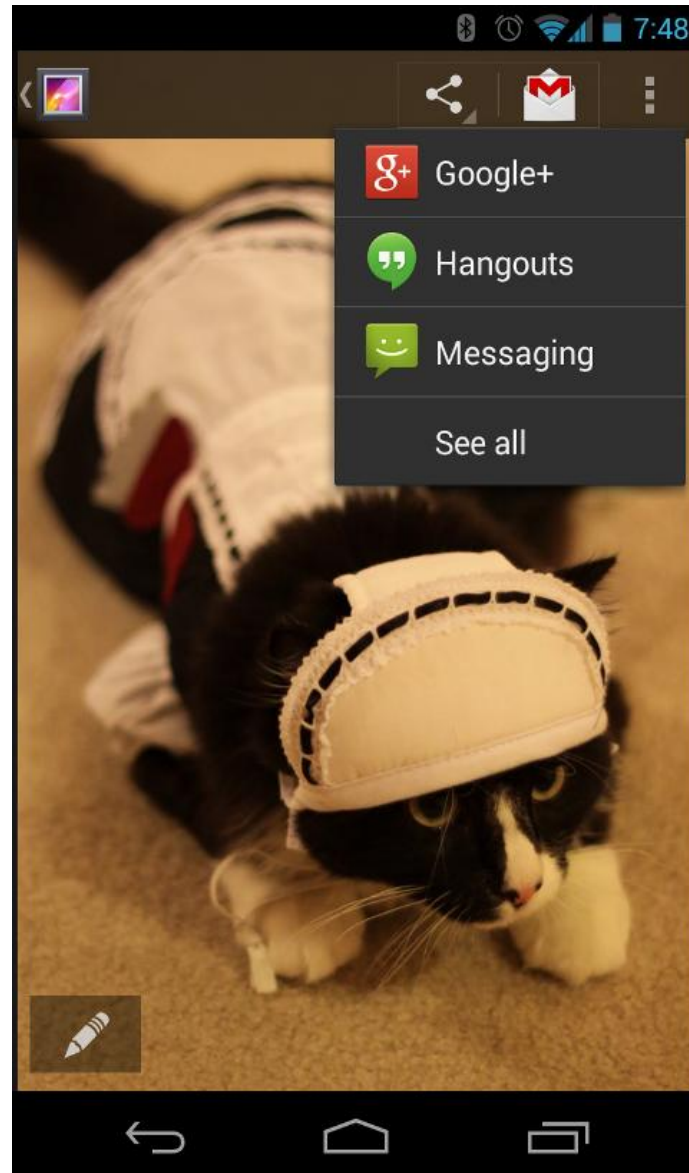
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnListview = (Button) findViewById(R.id.btn_listview);
        btnStyling = (Button) findViewById(R.id.btn_styling);
        btnDatabase = (Button) findViewById(R.id.btn_database);
        btnSharedPreferences = (Button) findViewById(R.id.btn_shared_prefs);
        btnWebview = (Button) findViewById(R.id.btn_webview);
        btnWebServices = (Button) findViewById(R.id.btn_web_services);
    }
}
```

The bottom toolbar includes tabs for Problems, Javadoc, Declaration, Console, and LogCat. The LogCat tab is selected and highlighted with a red box. It features a search bar with the text "Search for messages. Accepts Java regexes. Prefix with pid; app; tag; or text: to limit scope." and a dropdown menu set to "verbose". Below the search bar is a table of log messages.

L...	Time	PID	TID	Application	Tag	Text
W	04-05 09:59:2...	33	33		Zygote	Preloaded drawable resou _handle_horizontal.9.png)
W	04-05 09:59:2...	33	33		Zygote	Preloaded drawable resou _handle_vertical.9.png)
D	04-05 09:59:2...	33	33		dalvikvm	GC_EXPLICIT freed 477 ok
W	04-05 09:59:2...	33	33		Zygote	Preloaded drawable resou

Android App Basics



Android Apps Building Blocks

Apps Development Flow

Build Cycle

Directory Structure

Android Activity

Activity Lifecycle

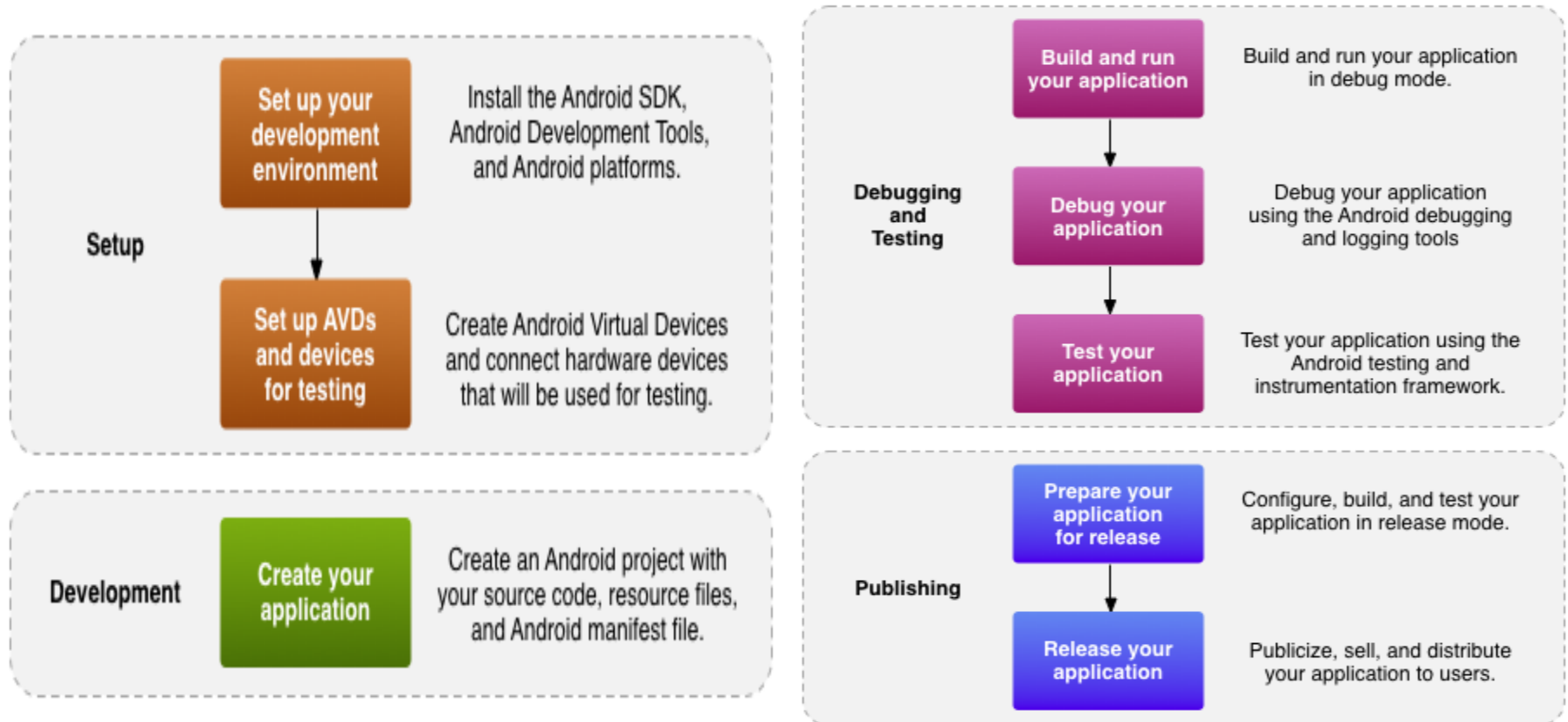
Android Manifest File / Android Permissions

Android Intent, Toasts

Android UI

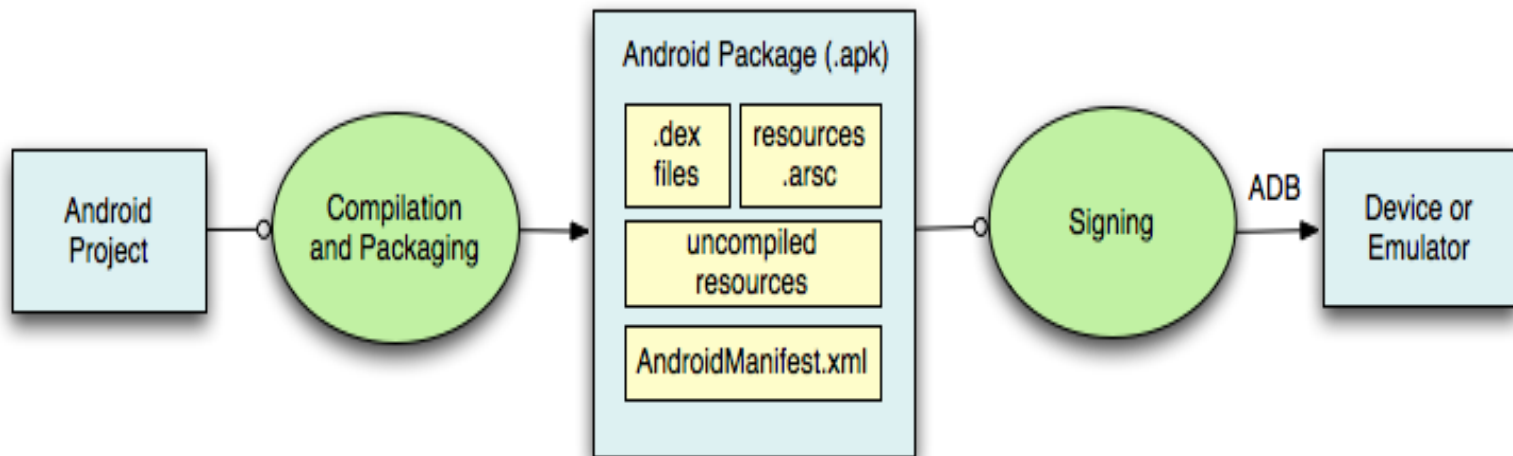
Other

Apps Development Flow

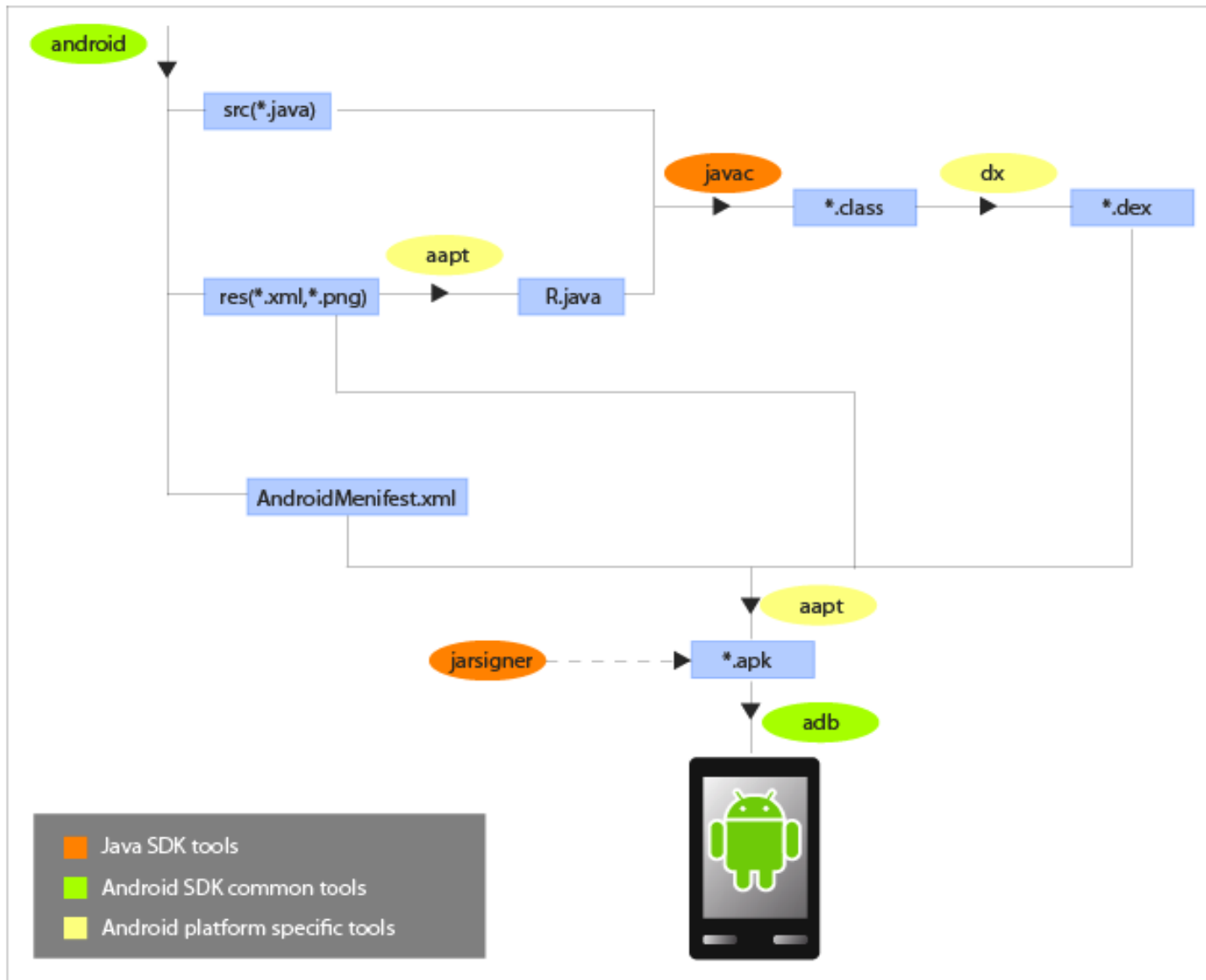


1. Setup 2. Development 3. Debugging & Testing 4. Publishing

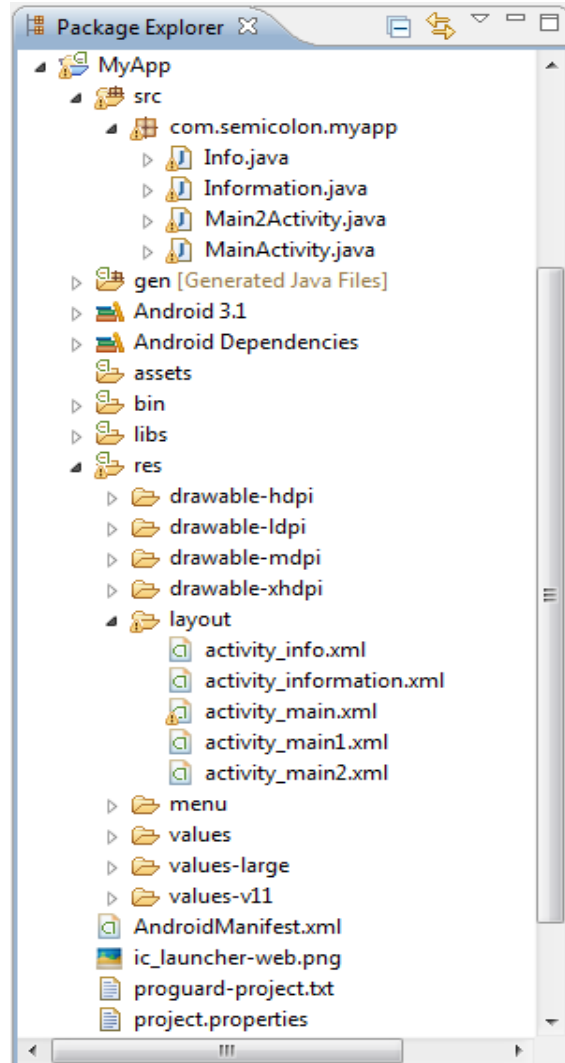
Apps Build Cycle



Apps Build Cycle



Android Apps Project – Directory Structure



src/

bin/

Output directory of the build. final .apk
gen/

R.java etc

res/

Contains application resources, such as
drawable files, layout files, and string
values.

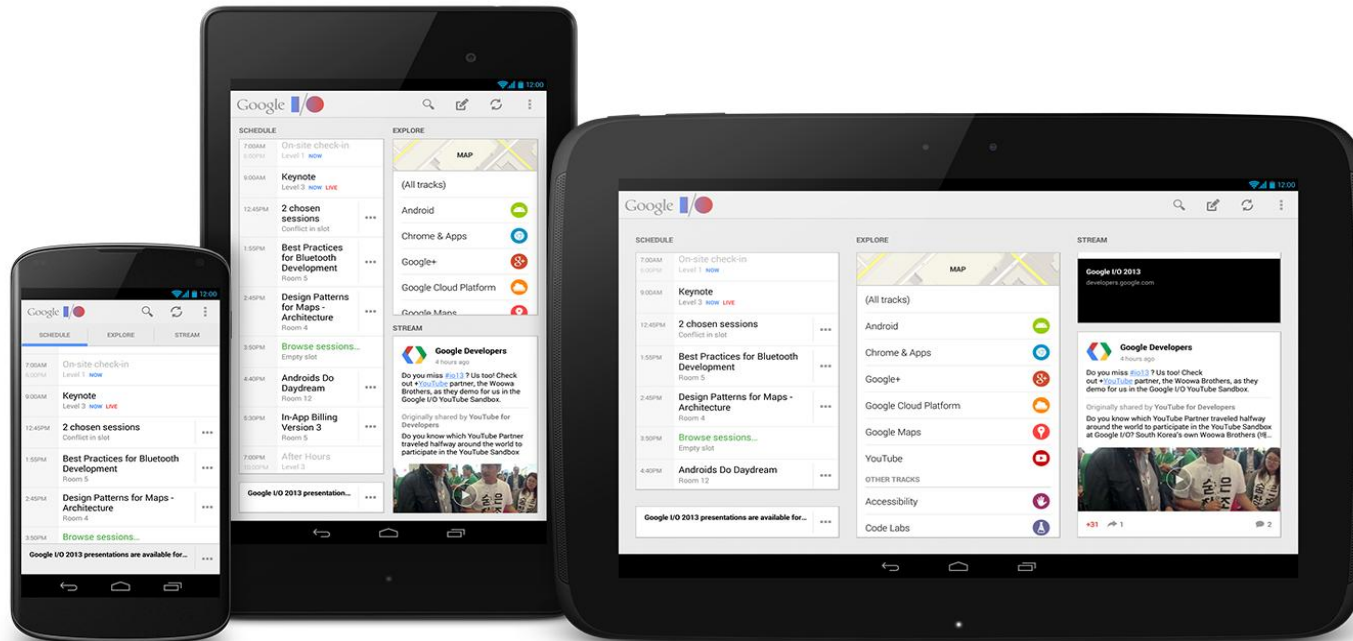
drawable/

layout/

libs/

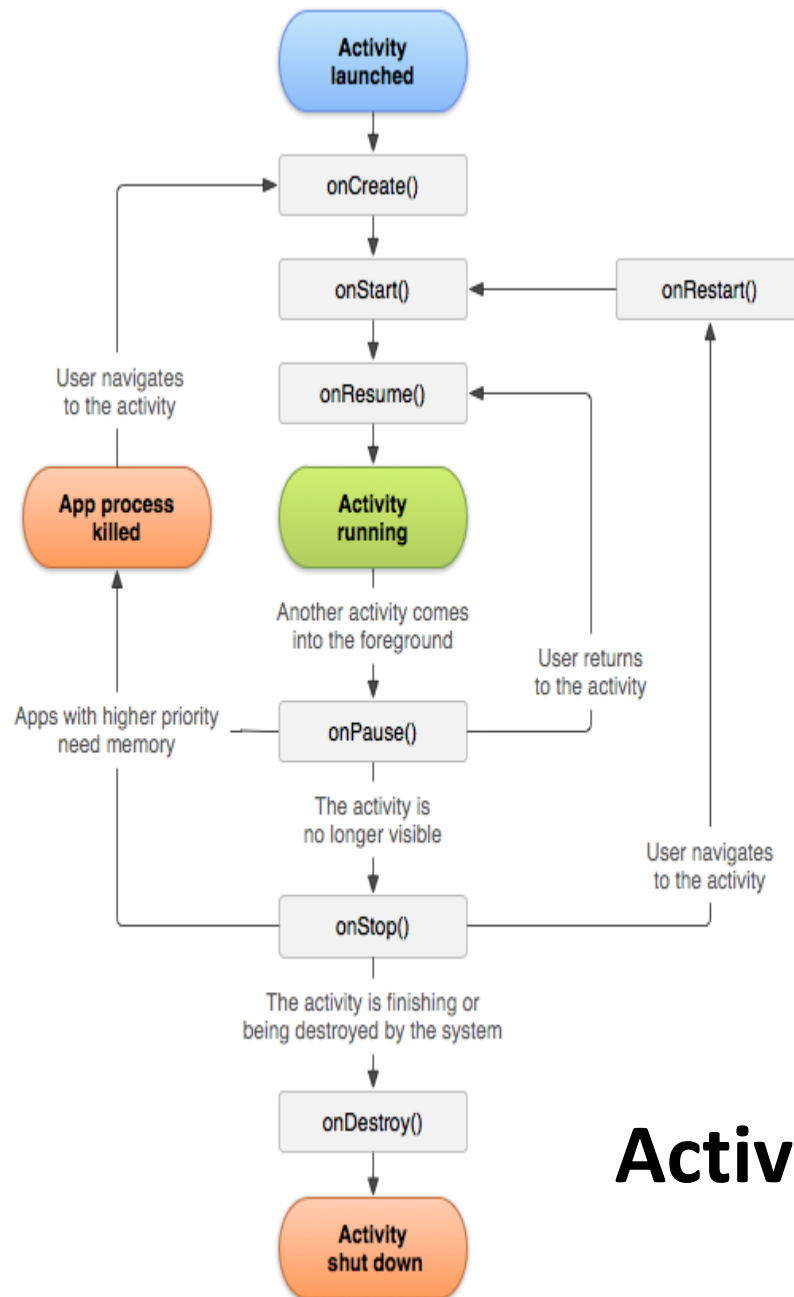
AndroidManifest.xml

Activity



Website – Pages ! App – Screens / Activities

Android Application = Σ activity



Activity Lifecycle

App Manifest File (AndroidManifest.xml)

App must have AndroidManifest.xml file
(with precisely that name) in its root directory.

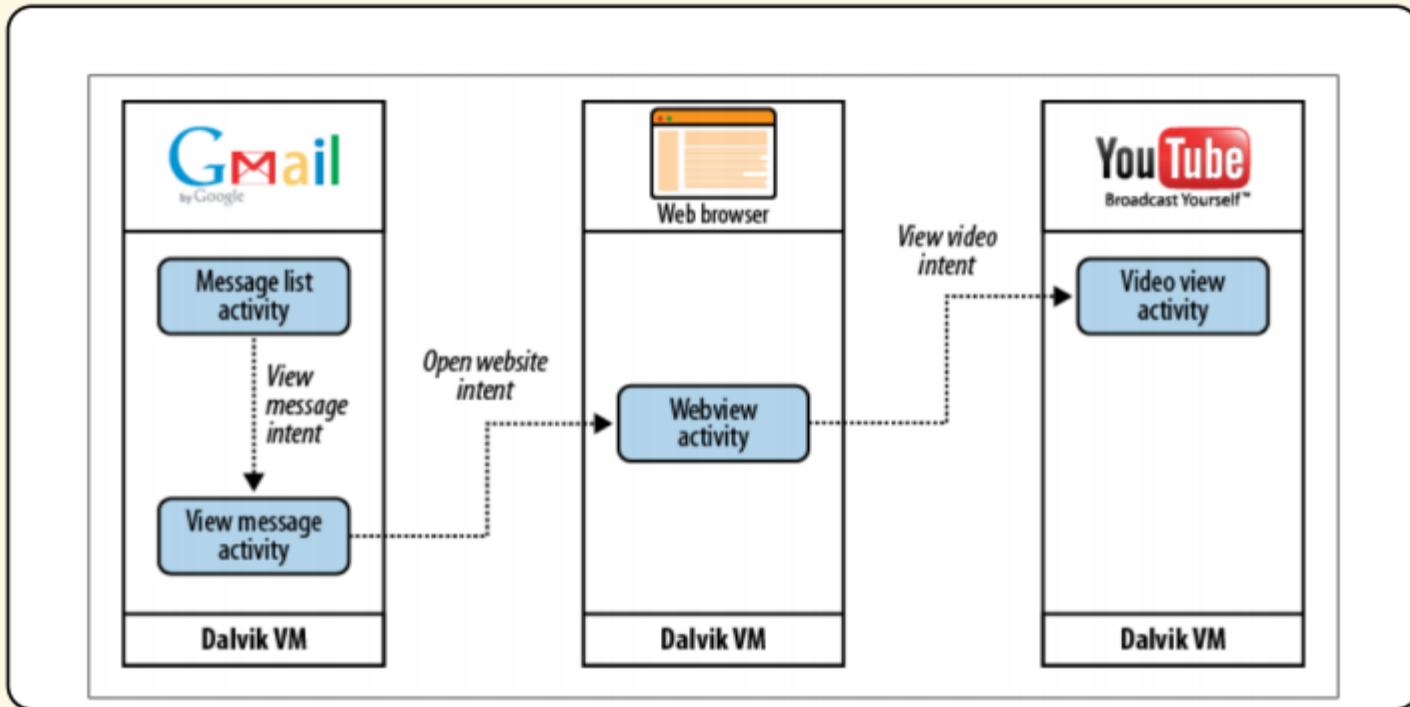
Java package for the application
(which serves as a unique identifier for the application)

Describe the components of the application
(activities, services, broadcast receivers, and content providers).

It declares permissions
(to access protected parts of API & interact with other apps)

declares minimum level Android API
(that the application requires)

Intent



Used to start activities, start/stop services, or send broadcasts

Android UI - Design



Build visually compelling apps that look great on any device.

Android UI Design : Learn Following

Understand Android Design Principals

(Enchant Me, Simplify My Life, Make Me Amazing)

Understand UI basics

(Basic building blocks and UI elements)

Understand UI Style Guidelines

(Different devices & displays, themes, metrics, typography, color, icon, branding, writing style etc)

Understand UI Design Patterns

(Dashboard, Action Bar, Navigation Drawer, Pull To Refresh)

Android Design Guide: <http://developer.android.com/design>

Android UI - Design

XML

Declare UI in XML
Inflate XML in JAVA files

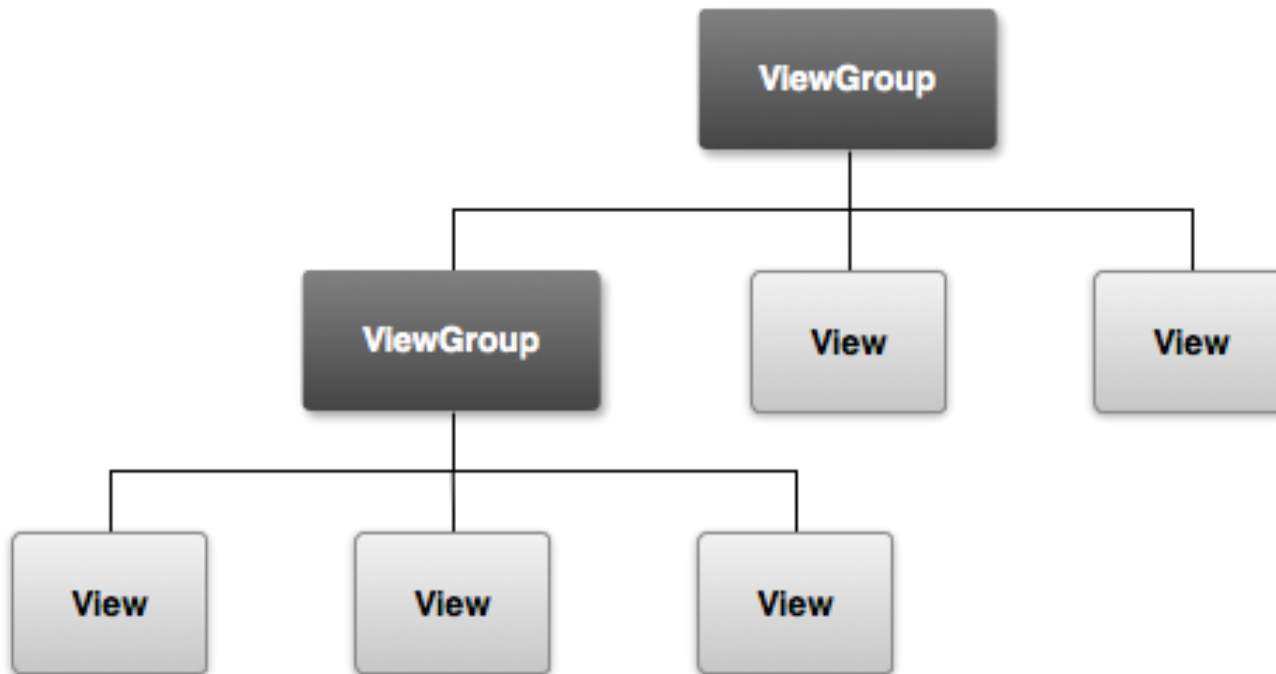
or

Programmatically

Initialize new widgets
Customize properties for each

use both methods as necessary

Android UI - Layout



View hierarchy which defines a UI layout

Android UI – Common Layouts

Linear Layout



A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

Relative Layout



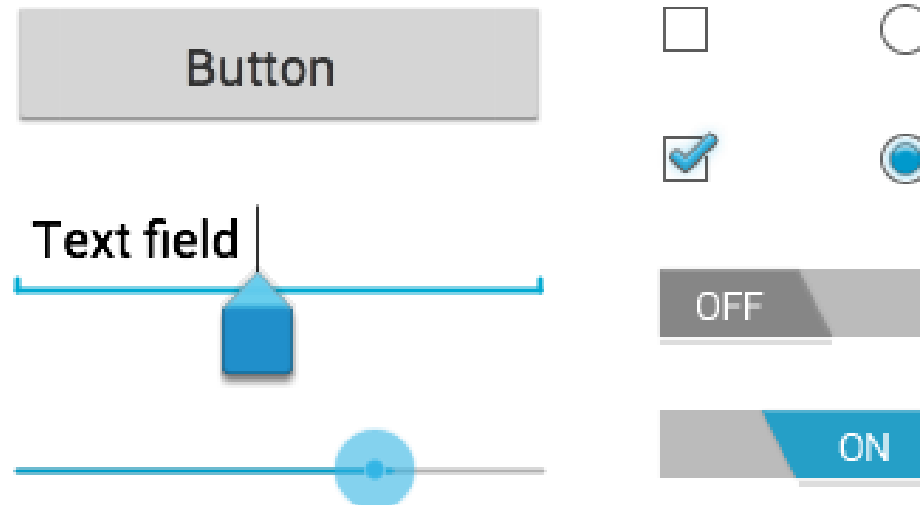
Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).

Web View

```
<html>  
  <!-- web page -->  
</html>
```

Displays web pages.

Android UI – Input Controls



```
<Button android:id="@+id/button_send"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/button_send"  
        android:onClick="sendMessage" />
```

Android UI – Components (Elements)

Layouts

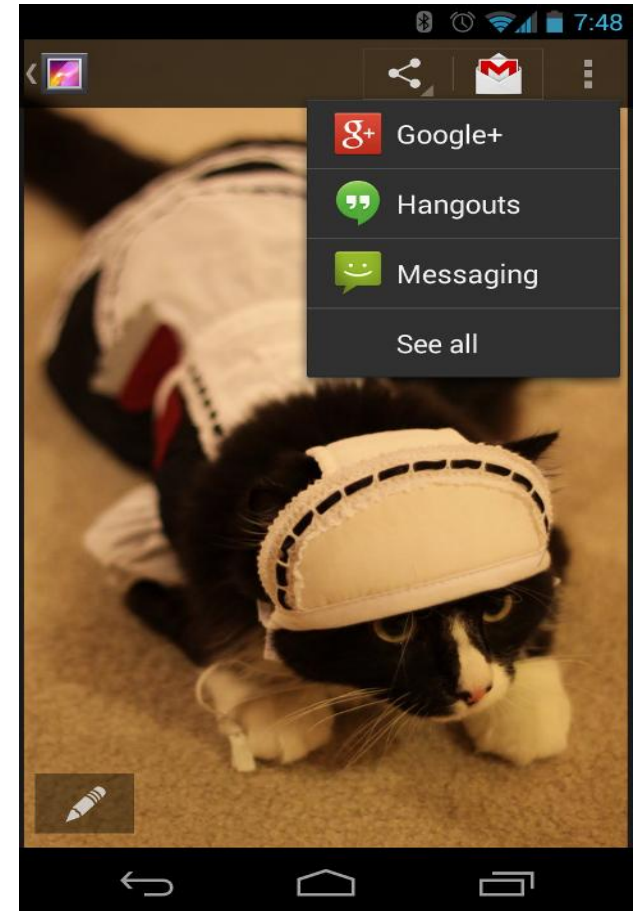
- Linear Layout
- Relative Layout
- List View
- Grid View

Input Controls

- Buttons
- Text Fields
- Checkboxes
- Radio Buttons
- Toggle Buttons
- Spinners
- Pickers

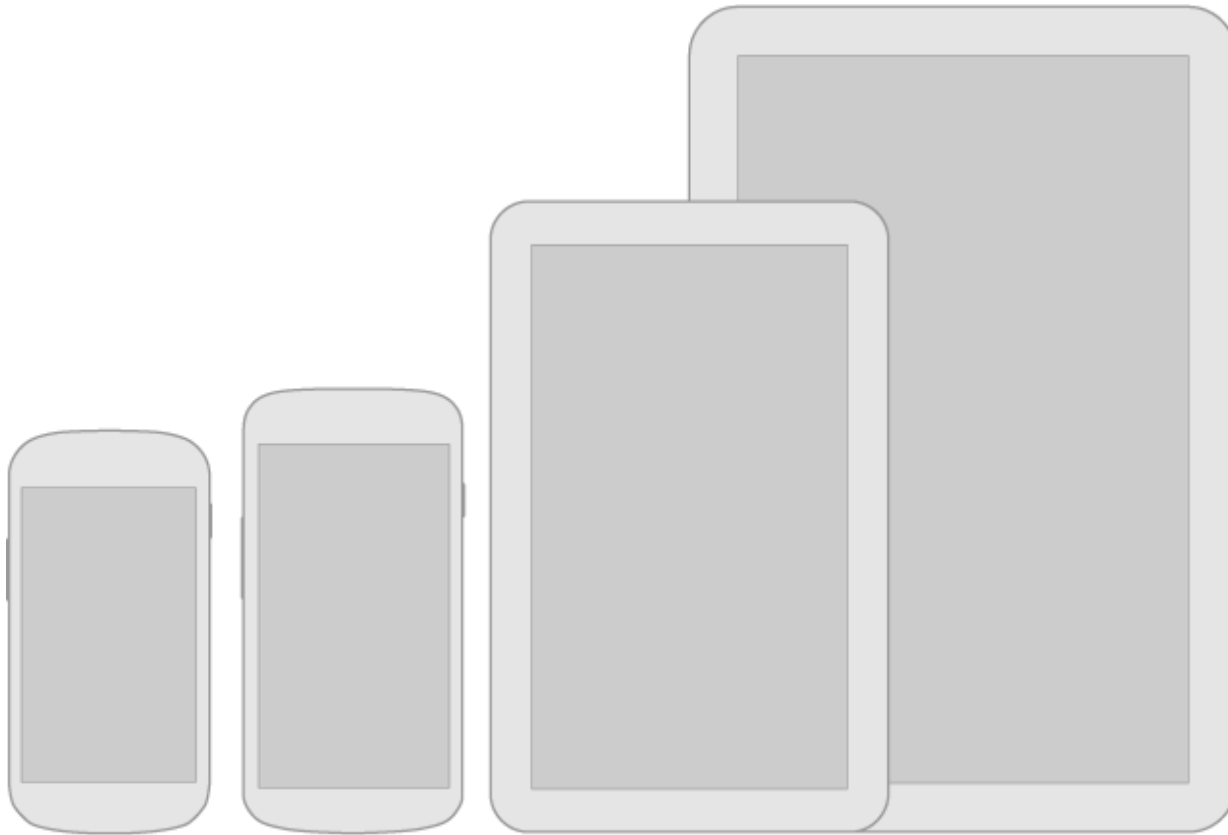
Input Events

- Menus
- Action Bar
- Settings
- Dialogs
- Notifications
- Toasts
- Search
- Drag and Drop
- Styles and Themes
- Custom Components



Full detail: <http://developer.android.com/guide/topics/ui>

Android UI - Different Devices & Displays



**Utilize full device screen, Reveal more content on larger devices,
provide resources for different device densities**

Android UI - Different Devices & Displays

Supporting different screen size & densities:

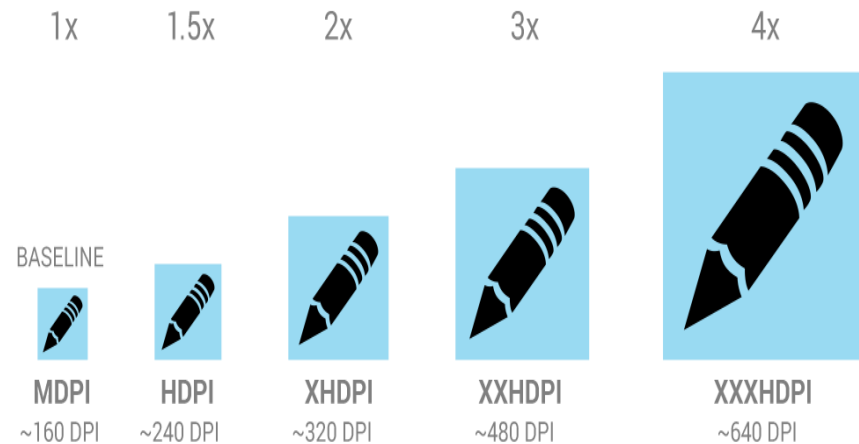
Start with Normal Size (MDPI) then scale up or down to build asset

Use Density Independent Pixels

DP for General Units
SP for Texts

Use "wrap_content" and "match_parent"

Use Nine-Patches



Data Storage (persistence storage)

Shared Preferences

(Store private primitive data in key-value pairs)

File Storage

(Store data on the device internal or external memory)

SQLite Databases

(Store structured data in a private database)

Network Connection

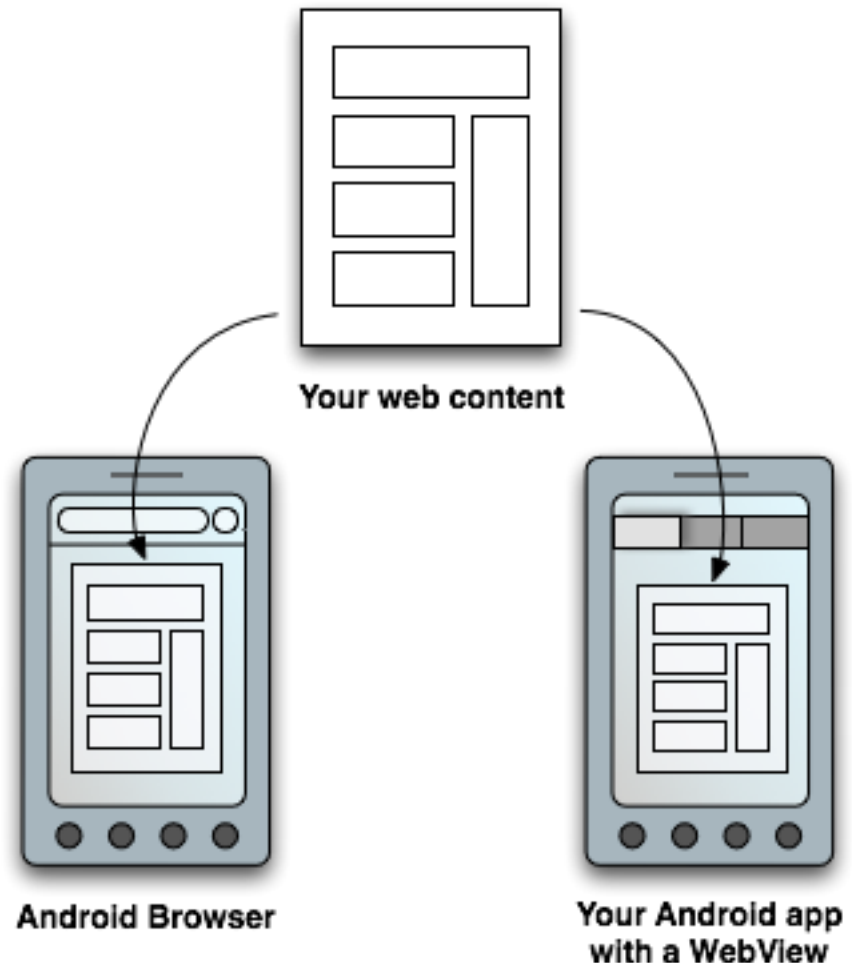
(Store data on the web with your own network server)



Web Apps with Android WebView

Make your web content available to users in two ways:

- 1) In a traditional web browser
- 2) In an Android application, by including a WebView in the layout.



Working with Web Services

Data Exchange between App and Server is done via JSON / XML etc

JSON (JavaScript Object Notation) is an independent data exchange format.

Data structures in JSON are based on key / value pairs.

JSON Object

JSON object is a set of key / value pairs which starts with "{" and ends with "}".

```
{  
  firstName: 'Lars',  
  lastName: 'Vogel',  
  address: {  
    street: 'Examplestr.',  
    number: '31'  
  }  
}
```


JSON Array

JSON Arrays are one or more values surrounded by [] and separated by ","

```
[
  {
    firstName: 'Lars',
    lastName: 'Vogel',
    address: {
      street: 'Examplestr.',
      number: '31'
    }
  },
  {
    firstName: 'Jack',
    lastName: 'Hack',
    address: {
      street: 'Examplestr.',
      number: '31'
    }
  }
]
```

Today's Workshop

Simple Dashboard App

Using ListView

Using Styling

Using Database (SQLite)

Using SharedPreferences

Using WebView

Using Web Services / JSON

What Next

App Components

Services

Content Providers

Broadcast Receivers

App Widgets

Processes & Threads

Other Important Topics

Media & Camera

Location and Sensors (GPS, Motion etc)

Connectivity (Wifi, Bluetooth, NFC)

UI Design Patterns

Third Party UI Libraries