Mobile Application Development

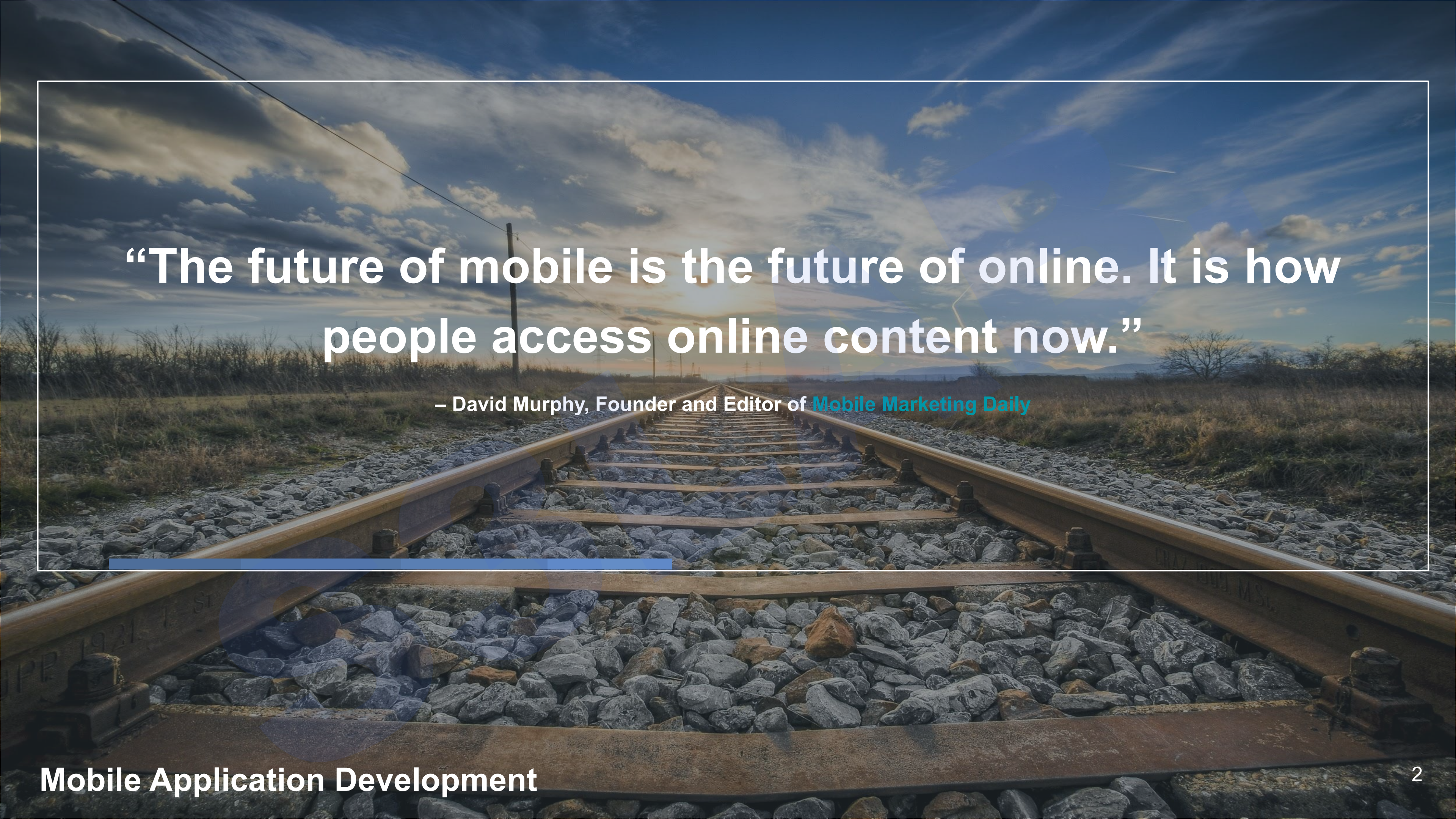
(Android + IDE + First Application)

Instructor: Thanh Binh Nguyen

February 1st, 2020

S^{3}Lab

Smart Software System Laboratory



> **“The future of mobile is the future of online. It is how people access online content now.”**
>
> **– David Murphy, Founder and Editor of Mobile Marketing Daily**

About Android

Overview

- A **Software Platform** and **OS** for **mobile**, **embedded** and **wearable** devices
- Based on the **Linux kernel** (2.6 kernel)
- **Google** is the **principle maintainer**
- Other companies contribute to the system.
- Allow writing **managed code** in the **java** language
- Each device manufacturer can **customize Android** to suit their needs
- An **Open Source** Project
- **Open Handset Alliance** Project

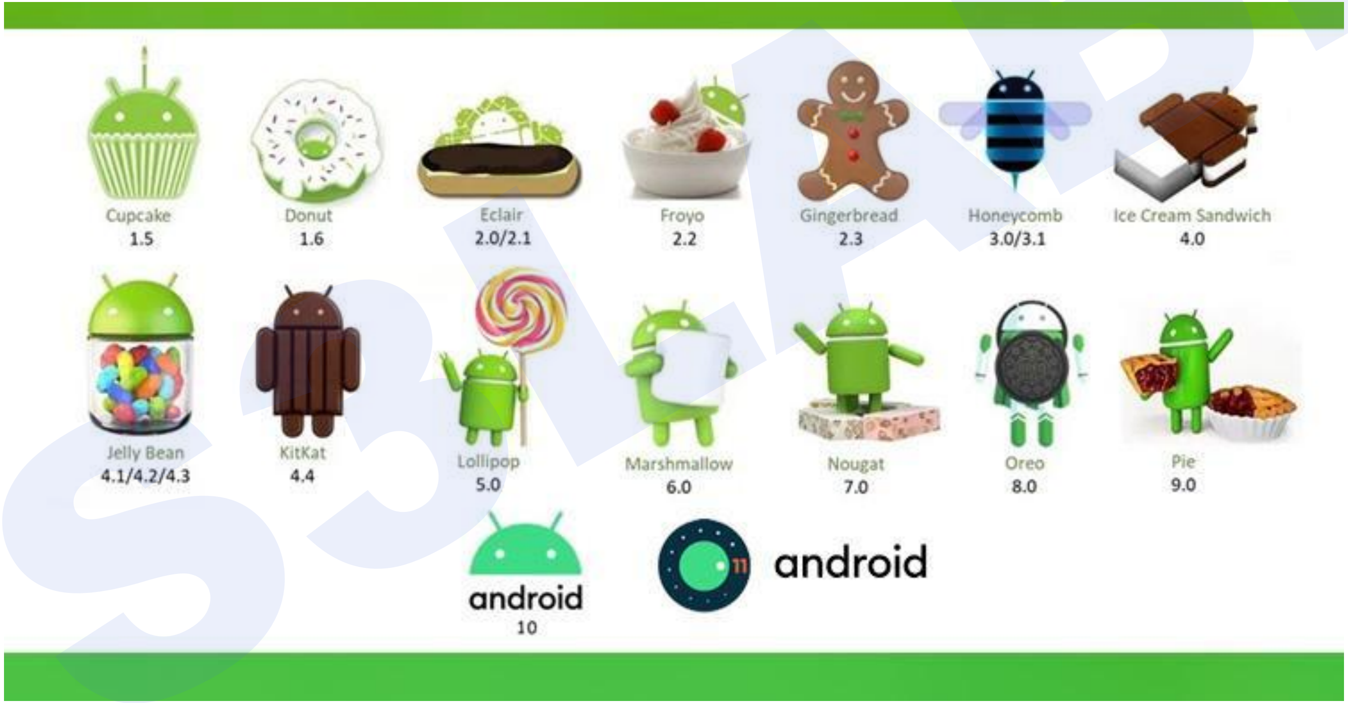
About Android

Who develop Android?

- Initially developed by **Andrew** (Andy) **Rubin** and his team in **Android Inc.**
- Google acquired Android Inc. in 2005.
- Till Mac 2013, developed by Google under Andy Rubin (Senior Vice President of Mobile)

About Android

From the past to ...



About Android

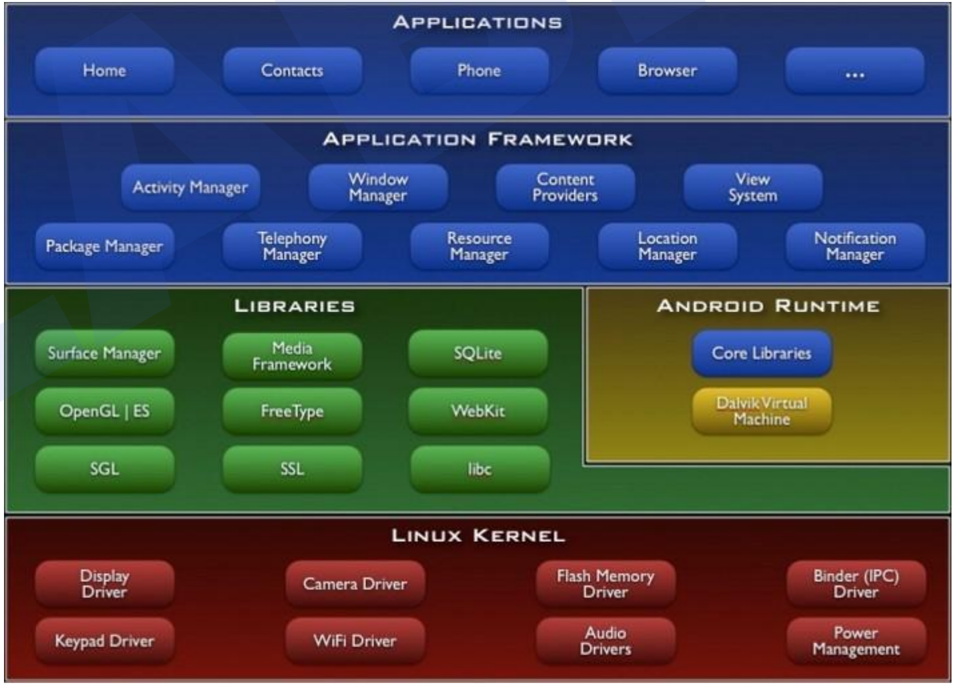
Architecture

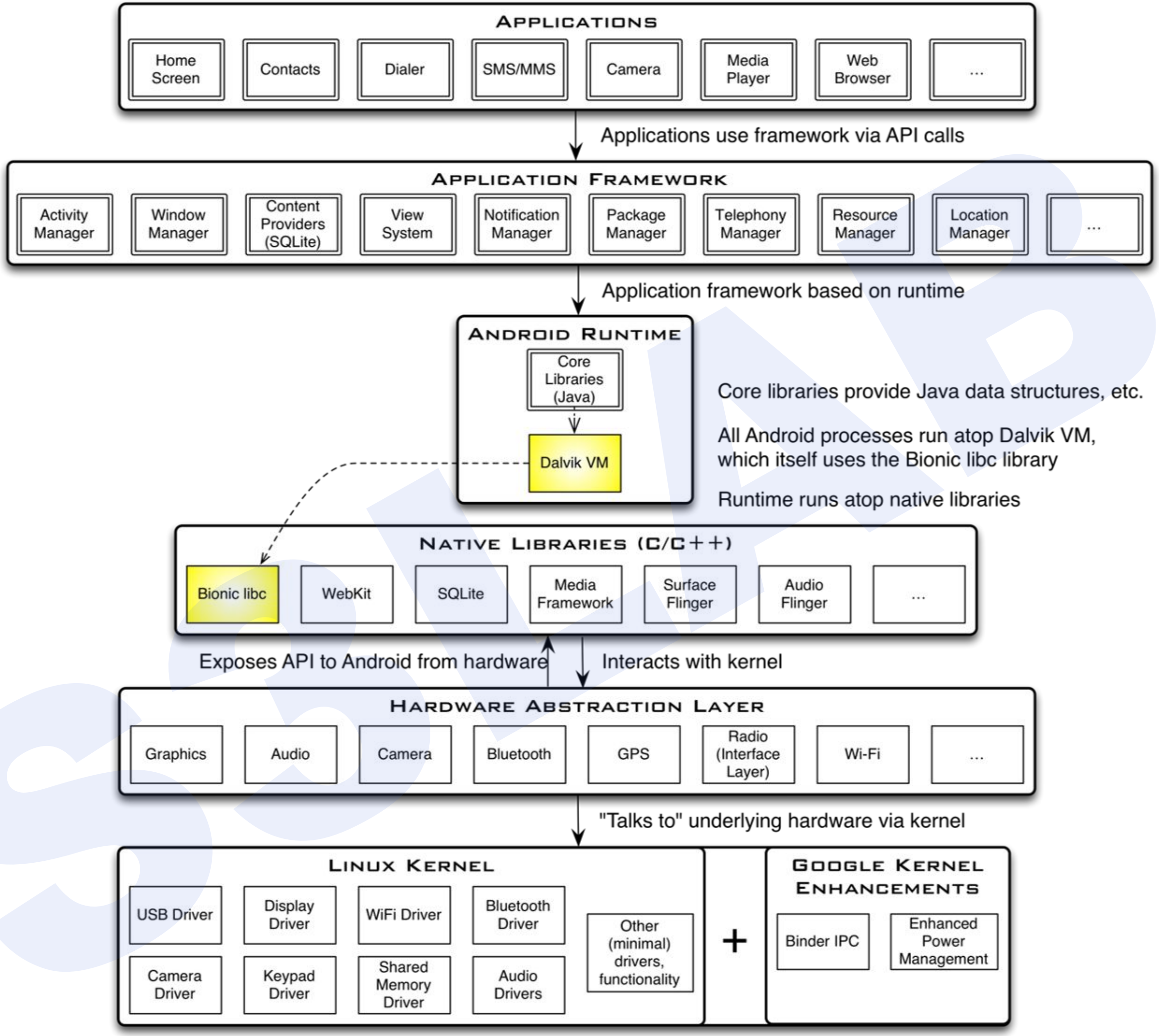
User applications
Use Java framework and, optionally, native code.

Android framework
Java classes under com.android

Native framework layer
User mode C, C++ code - compiled to native platform or 32bit compatibility mode on 64 bits.

Linux Kernel (GPL license)
C code - compiled to native platform (x86, arm, mips)





APPLICATIONS

Home Screen | Contacts | Dialer | SMS/MMS | Camera | Media Player | Web Browser | …

Applications use framework via API calls

APPLICATION FRAMEWORK

Activity Manager | Window Manager | Content Providers (SQLite) | View System | Notification Manager | Package Manager | Telephony Manager | Resource Manager | Location Manager | …

Application framework based on runtime

ANDROID RUNTIME

Core Libraries (Java)

Dalvik VM

Core libraries provide Java data structures, etc.

All Android processes run atop Dalvik VM, which itself uses the Bionic libc library

Runtime runs atop native libraries

NATIVE LIBRARIES (C/C++)

Bionic libc | WebKit | SQLite | Media Framework | Surface Flinger | Audio Flinger | …

Exposes API to Android from hardware

Interacts with kernel

HARDWARE ABSTRACTION LAYER

Graphics | Audio | Camera | Bluetooth | GPS | Radio (Interface Layer) | Wi-Fi | …

"Talks to" underlying hardware via kernel

LINUX KERNEL

USB Driver | Display Driver | WiFi Driver | Bluetooth Driver | Camera Driver | Keypad Driver | Shared Memory Driver | Audio Drivers | Other (minimal) drivers, functionality

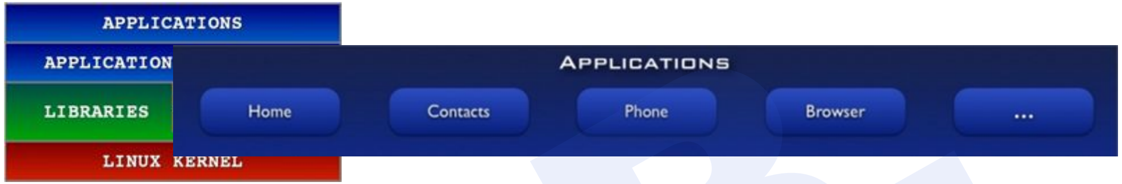
+

GOOGLE KERNEL ENHANCEMENTS

Binder IPC | Enhanced Power Management

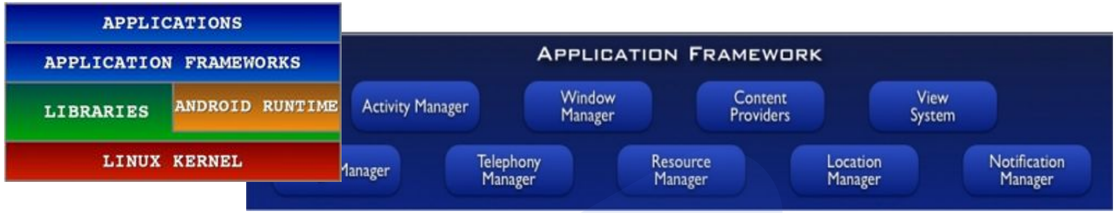
About Android

Architecture - Application



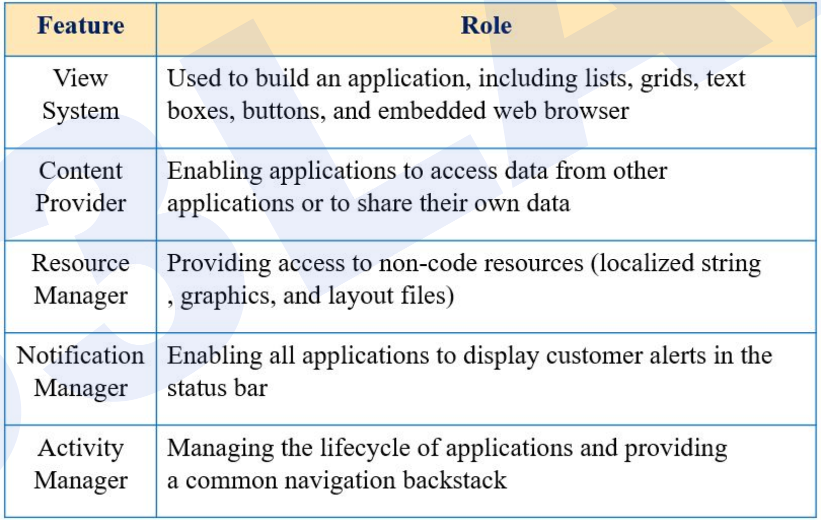
- Android provides a set of core applications:
  - Email Client
  - SMS Program
  - Calendar
  - Maps
  - Browser
  - Contacts
  - Etc
- All applications are written using the Java language

About Android



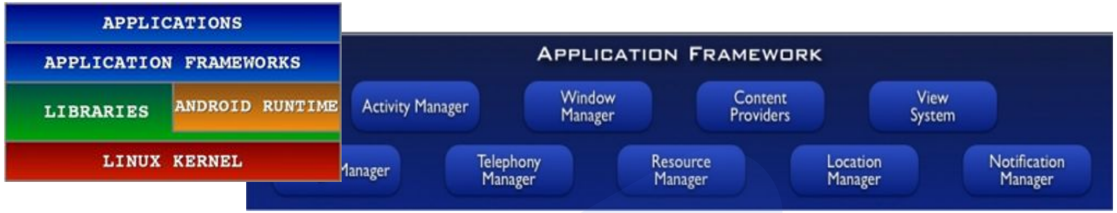
Architecture - Application Framework

- Most of the application framework accesses these core libraries through the Dalvik VM, the gateway to the Android Platform



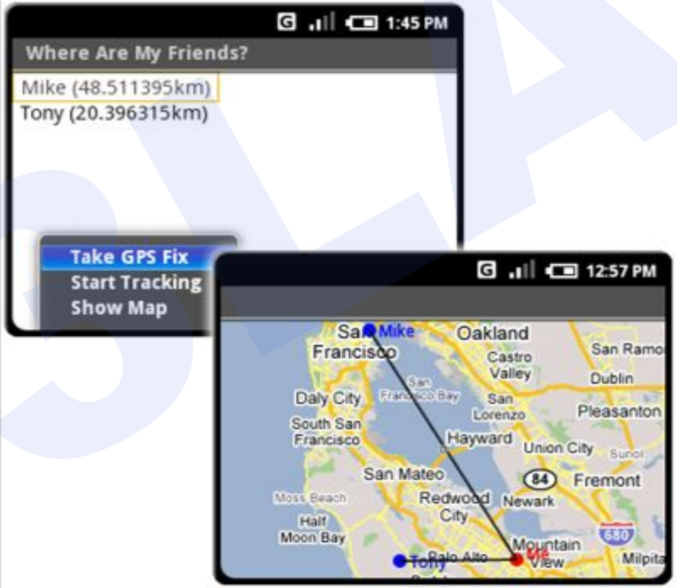
Feature	Role
View System	Used to build an application, including lists, grids, text boxes, buttons, and embedded web browser
Content Provider	Enabling applications to access data from other applications or to share their own data
Resource Manager	Providing access to non-code resources (localized string , 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

About Android



Architecture - Application Framework

- Location Manager



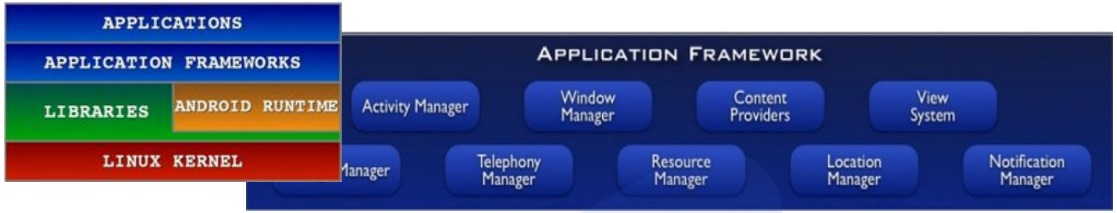
Where Are My Friends?

Mike (48.511395km)

Tony (20.396315km)

About Android

Architecture - Application Framework



APPLICATIONS

APPLICATION FRAMEWORKS

LIBRARIES | ANDROID RUNTIME

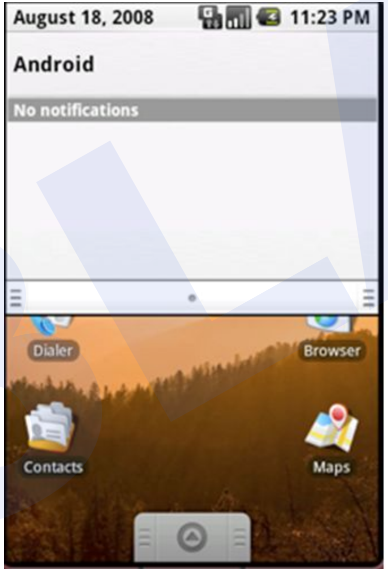
LINUX KERNEL

APPLICATION FRAMEWORK

Activity Manager · Window Manager · Content Providers · View System

Manager · Telephony Manager · Resource Manager · Location Manager · Notification Manager

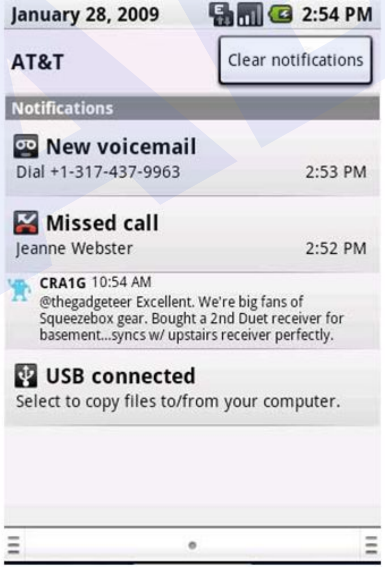
- Notification Manager
  - ○ How background app Interact with users
  - ○ Consistent notification presentation



August 18, 2008 11:23 PM

Android

No notifications



January 28, 2009 2:54 PM

AT&T

Clear notifications

Notifications

New voicemail
Dial +1-317-437-9963 2:53 PM

Missed call
Jeanne Webster 2:52 PM

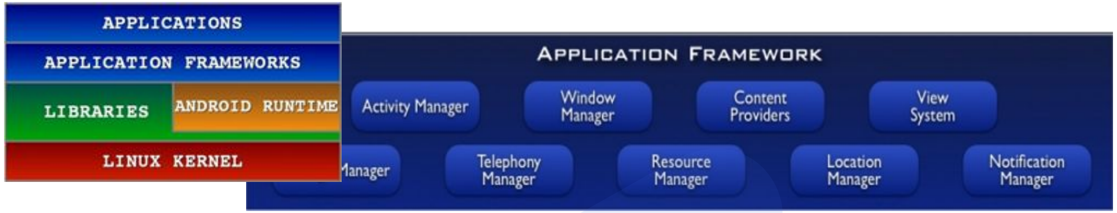
CRA1G 10:54 AM
@thegadgeteer Excellent. We're big fans of Squeezebox gear. Bought a 2nd Duet receiver for basement...syncs w/ upstairs receiver perfectly.

USB connected
Select to copy files to/from your computer.

XMPP

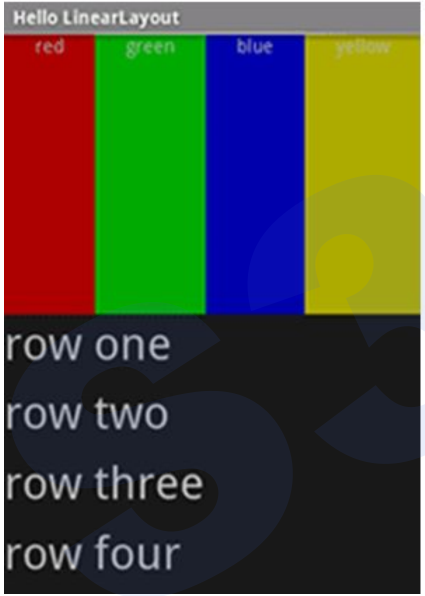
Sample Notification

About Android



Architecture - Application Framework

- View System

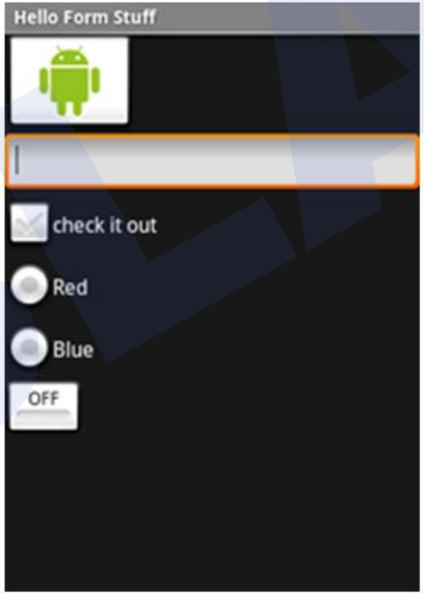

Hello LinearLayout

row one

row two

row three

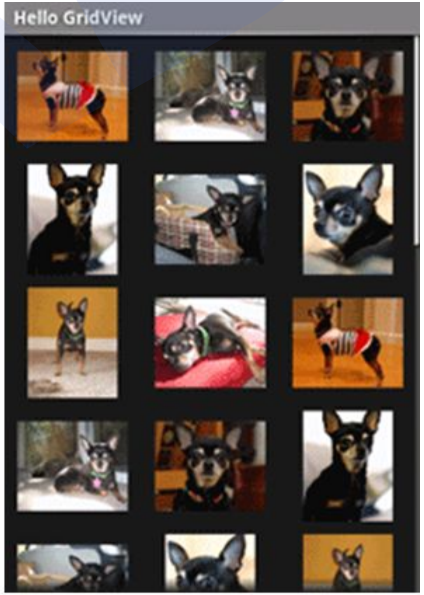
row four


Hello Form Stuff

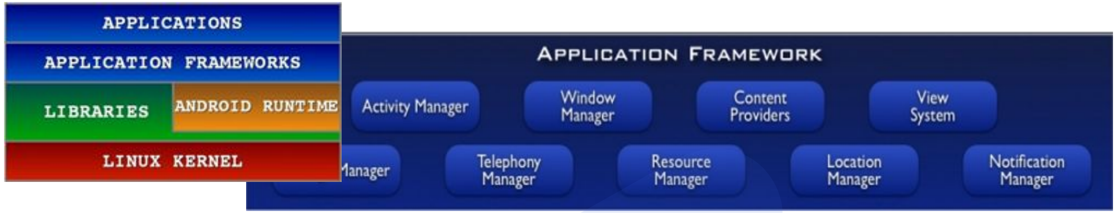
□ check it out

○ Red

○ Blue

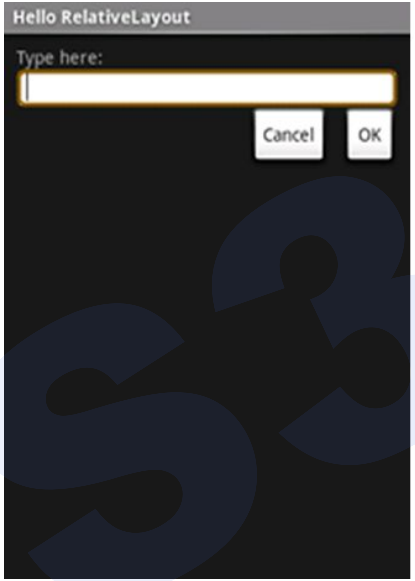

Hello GridView

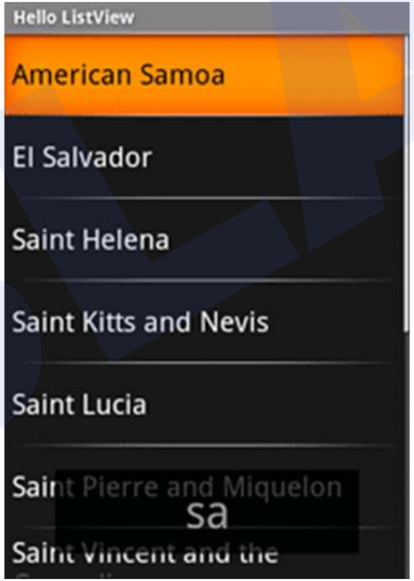
About Android

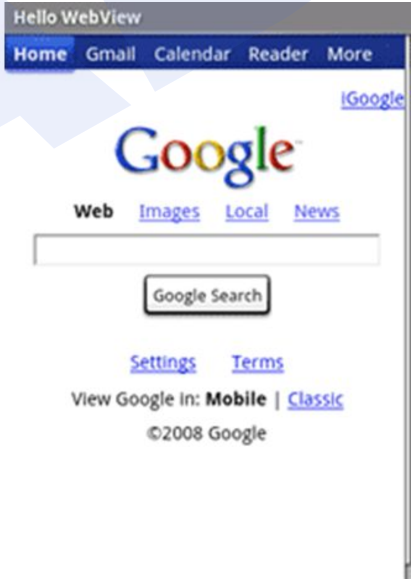


Architecture - Application Framework

- View System

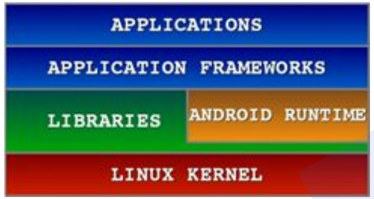


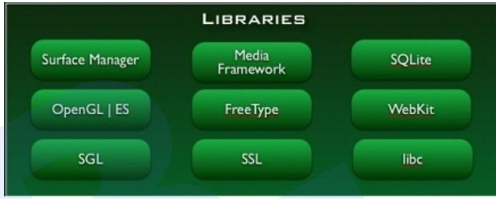




About Android

Architecture - Libraries

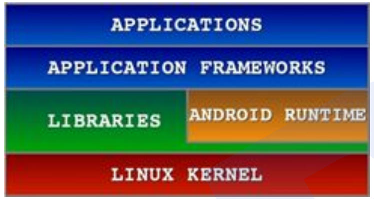


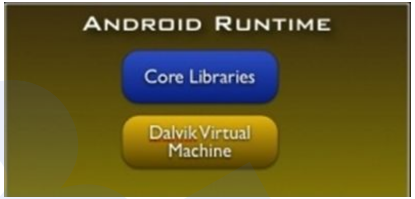


- Including a set of C/C++ libraries used by components of the Android system
- Exposed to developers through the Android application framework
  - Media -> PacketVideo's OpenCORE for recording, playback audio and video
  - Surface Manager -> controls access to the display system and supports 2D, 3D
  - WebKit -> for browser support.
  - FreeType -> font support
  - SQLite -> a relational database

About Android

Architecture - Android Runtime

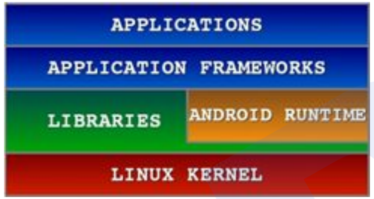


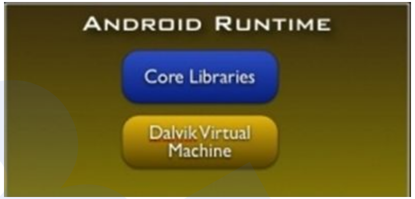


- Core Libraries
  - Providing most of the functionality available in the core libraries of the Java language
  - APIs
    - Data Structures
    - Utilities
    - File Access
    - Network Access
    - Graphics
    - Etc

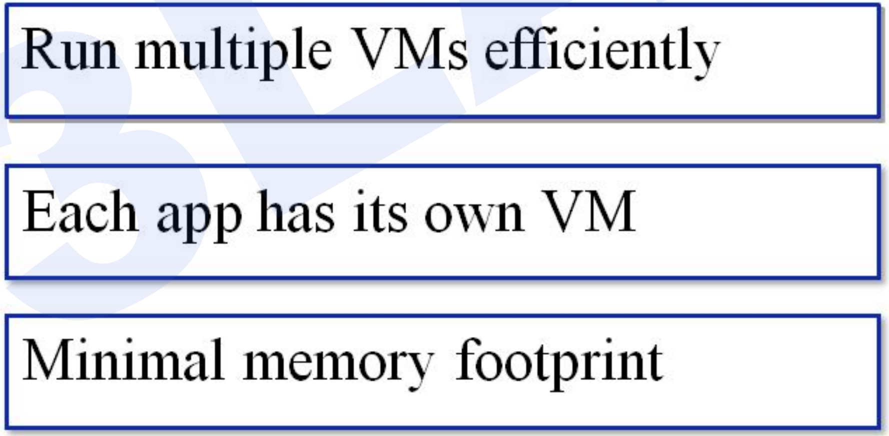
About Android

Architecture - Android Runtime





- Dalvik Virtual Machine



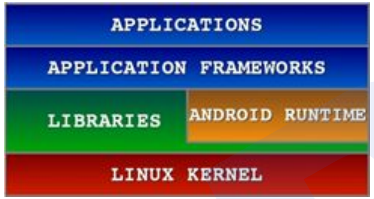
Run multiple VMs efficiently

Each app has its own VM

Minimal memory footprint

About Android

Architecture - Android Runtime



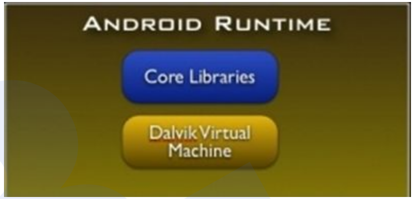
APPLICATIONS

APPLICATION FRAMEWORKS

LIBRARIES

ANDROID RUNTIME

LINUX KERNEL

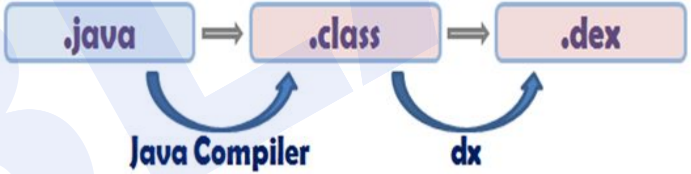


ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

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



.java ⟹ .class ⟹ .dex

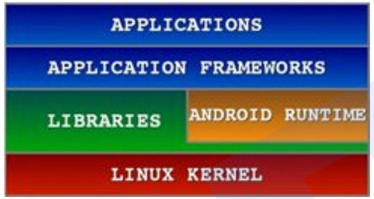
Java Compiler

dx

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

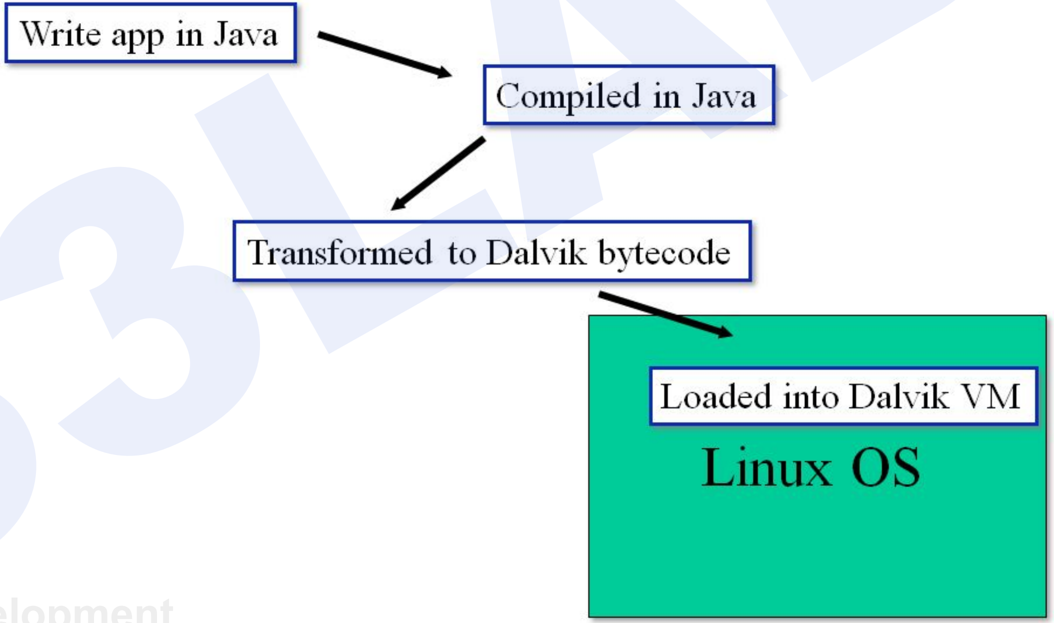
About Android

Architecture - Android Runtime



Android Runtime
Core Libraries
Dalvik Virtual Machine

- Dalvik Virtual Machine



Write app in Java

Compiled in Java

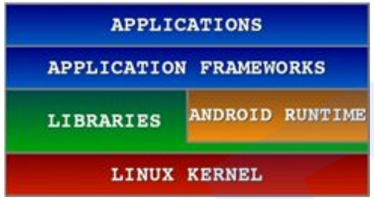
Transformed to Dalvik bytecode

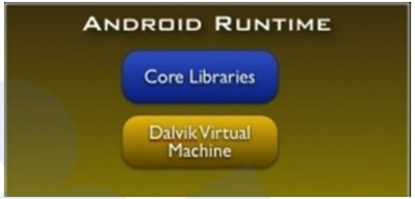
Loaded into Dalvik VM

Linux OS

About Android

Architecture - Android Runtime

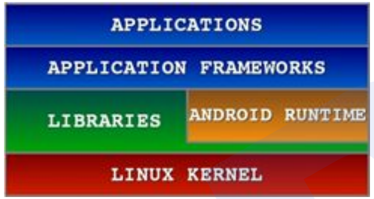




- From KITKAT, Dalvik Virtual Machine was replaced by ART
- ART introduces ahead-of-time (AOT) compilation, which can improve app performance. ART also has tighter install-time verification than Dalvik.

About Android

Architecture - Android Runtime



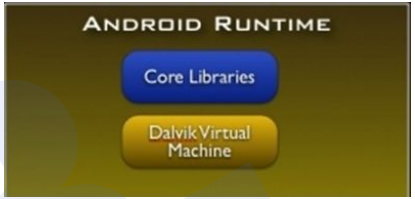
APPLICATIONS

APPLICATION FRAMEWORKS

LIBRARIES

ANDROID RUNTIME

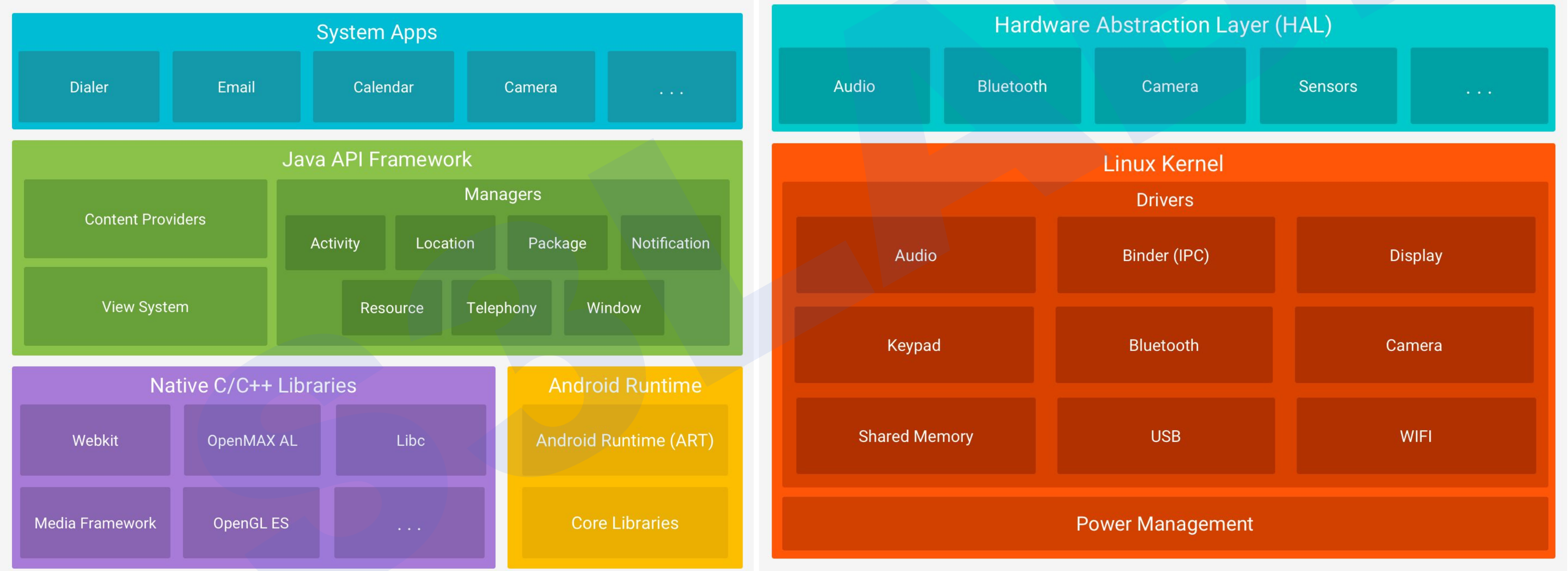
LINUX KERNEL



ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine



System Apps

Dialer

Email

Calendar

Camera

. . .

Java API Framework

Content Providers

View System

Managers

Activity

Location

Package

Notification

Resource

Telephony

Window

Native C/C++ Libraries

Webkit

OpenMAX AL

Libc

Media Framework

OpenGL ES

. . .

Android Runtime

Android Runtime (ART)

Core Libraries

Hardware Abstraction Layer (HAL)

Audio

Bluetooth

Camera

Sensors

. . .

Linux Kernel

Drivers

Audio

Binder (IPC)

Display

Keypad

Bluetooth

Camera

Shared Memory

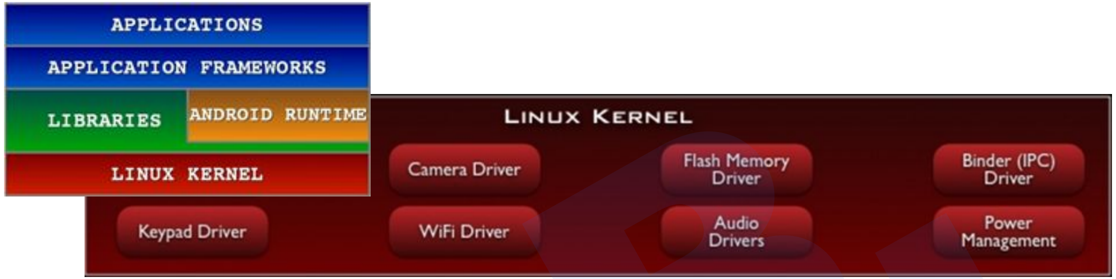
USB

WIFI

Power Management

About Android

Architecture - Linux Kernel



- Relying on Linux Kernel 2.6 for core system services
  - Memory and Process Management
  - Network Stack
  - Driver Model
  - Security
- The supplied device drivers include Display, Camera, Keypad, WiFi, Flash Memory, Audio, and IPC (interprocess communication).
- Providing an abstraction layer between the H/W and the rest of the S/W stack

About Android

versioning

- Platform version
  - Current one is 11
- Framework API level
  - SDK compatibility
  - Each platform version has an API level
- NDK API level
  - API level for native headers
- Distribution
  - http://developer.android.com/about/dashboards/index.html

About Android

Browsing the android source

- Source at
  - https://android.googlesource.com/
- Porting instructions (for system developers)
  - https://source.android.com/devices/index.html
- com.android classes
  - http://developer.android.com/reference/packages.html

Android Application Development

- Android SDK
  - Provides the Java framework classes
  - Compiles to java bytecode
  - Class framework is updated with every OS release
- Android NDK
  - C/C++ toolchain for compiling to machine code
- Android platform tools
  - adb (android debug bridge) : runs and debugs apps from your dev machine
- Android developer tools
  - Eclipse plug-in for Android
  - Android studio

Android Application Development

Application Packages

- .apk files: compressed files
  - class byte code
  - resources(icons, sounds, etc).
  - Binary native files
- All .apks are signed
  - Default development key is created by SDK.
  - When updating an application, signature are checked.

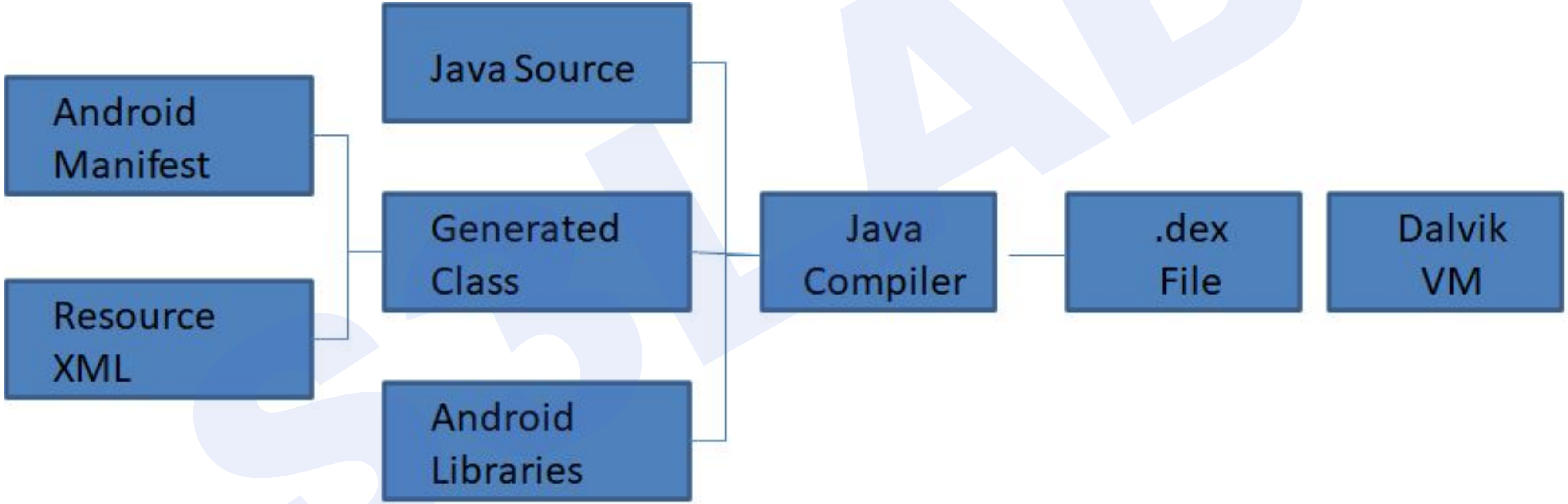
Android Application Development

Installing an application

- From application distribution markets
  - Google Play
  - Amazon App Store
  - Samsung App Store
  - Or your own distribution channel
- From your local computer using adb

Android Application Development

Compiling



Android Application Development

Core Building Blocks (Fundamental Components)

- Activity
- Views
- Fragments
- Intents
- Services
- Content Provider
- Broadcast Receivers
- Android Manifest.xml

Android Application Development

Core Building Blocks - Activity, View and Fragment

- An **Activity** is a class that represents to one UI screen
- A **View** is the UI element such as button, label, text field etc. Anything that you see is a view.
- **Fragments** are like parts of activity. An activity can display one or more fragments on the screen at the same time.

Android Application Development

Core Building Blocks - Intent

- **Intent** is used to invoke components. It is mainly used to:
  - Start the service
  - Launch an activity
  - Display a web page
  - Display a list of contacts
  - Broadcast a message
  - Dial a phone call etc.

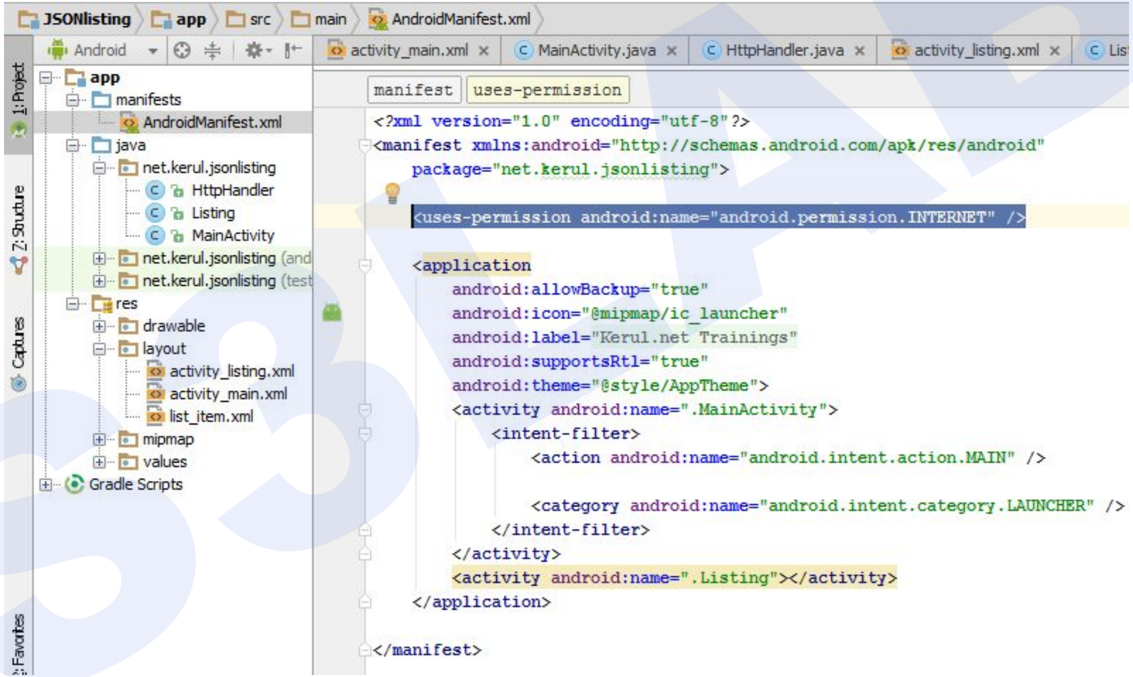
Android Application Development

Core Building Blocks - Services, Broadcast Receivers, AndroidManifest.xml

- **Services**: Faceless components that run in the background
  - E.g. music player, network download etc…
- **AndroidManifest.xml**: It contains informations about activities, content providers, permissions etc. It is like the web.xml file in Java EE.
- **Broadcast Receivers**: handle communication between core Android OS and applications running on the surface layer. The application modules communicate with each other using the broadcast receiver.

Android Application Development

Core Building Blocks - Services, Broadcast Receivers, AndroidManifest.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.kerul.jsonlisting">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Kerul.net Trainings"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Listing"></activity>
    </application>

</manifest>
```

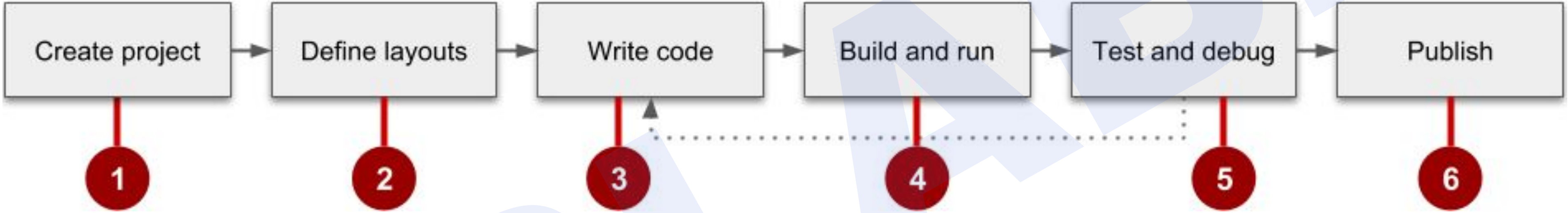
Android Application Development

Core Building Blocks - Content Providers

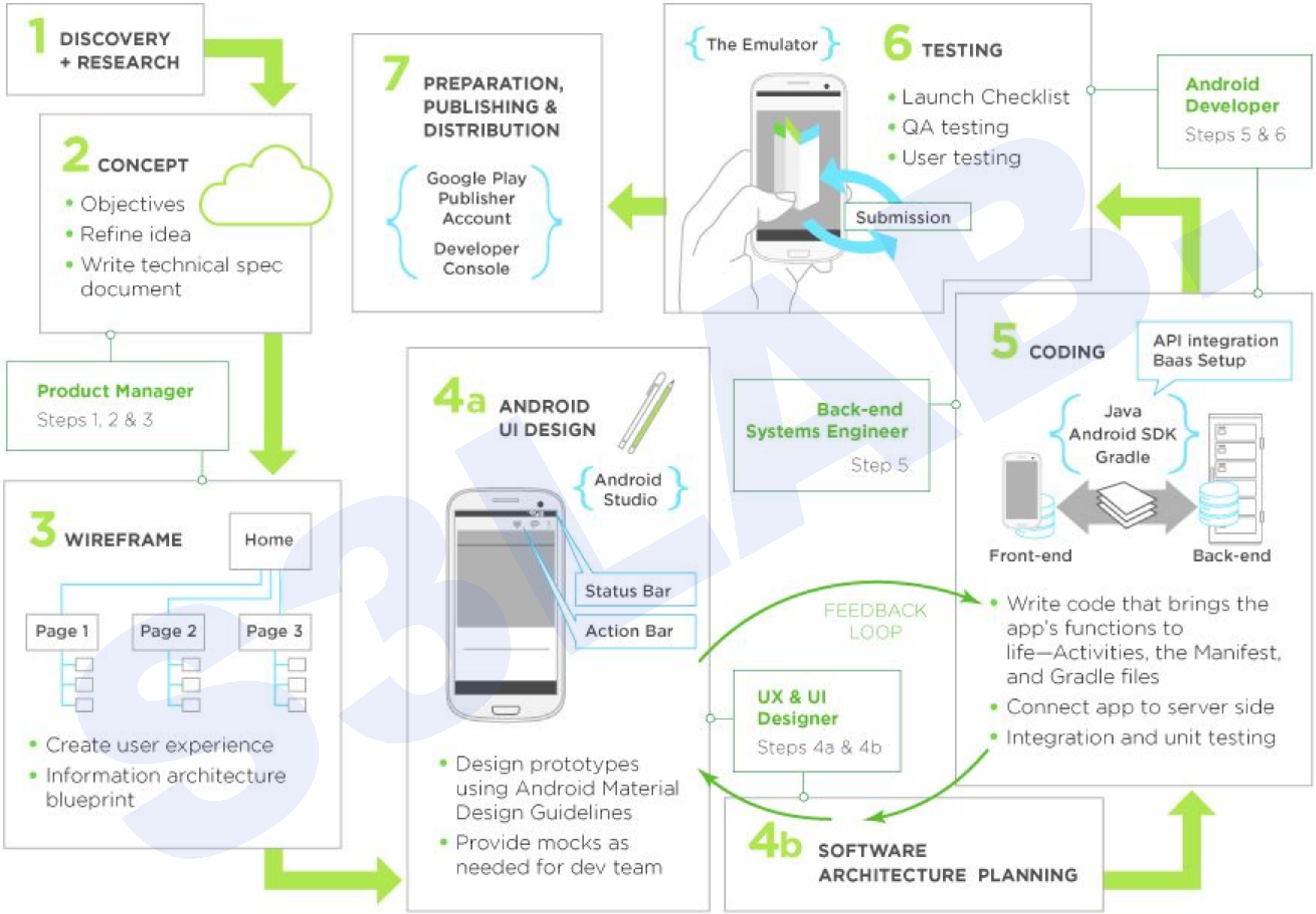
- Enables sharing of data across applications
  - E.g. address book, photo gallery
- Provides uniform APIs for:
  - Querying
  - delete, update and insert.
- Content is represented by URI and MIME type

Android Application Development

Development Process



- Layouts in XML format
- Build and run the app on real or virtual devices
- Publish the app by assembling the final APK and distributing it through channels such as Google play



1 DISCOVERY + RESEARCH

2 CONCEPT

- Objectives
- Refine idea
- Write technical spec document

Product Manager
Steps 1, 2 & 3

3 WIREFRAME

Home

Page 1 Page 2 Page 3

- Create user experience
- Information architecture blueprint

4a ANDROID UI DESIGN

{ Android Studio }

Status Bar

Action Bar

- Design prototypes using Android Material Design Guidelines
- Provide mocks as needed for dev team

UX & UI Designer
Steps 4a & 4b

FEEDBACK LOOP

4b SOFTWARE ARCHITECTURE PLANNING

5 CODING

API integration Baas Setup

{ Java Android SDK Gradle }

Front-end

Back-end

- Write code that brings the app's functions to life—Activities, the Manifest, and Gradle files
- Connect app to server side
- Integration and unit testing

Back-end Systems Engineer
Step 5

Android Developer
Steps 5 & 6

{ The Emulator }

6 TESTING

- Launch Checklist
- QA testing
- User testing

Submission

7 PREPARATION, PUBLISHING & DISTRIBUTION

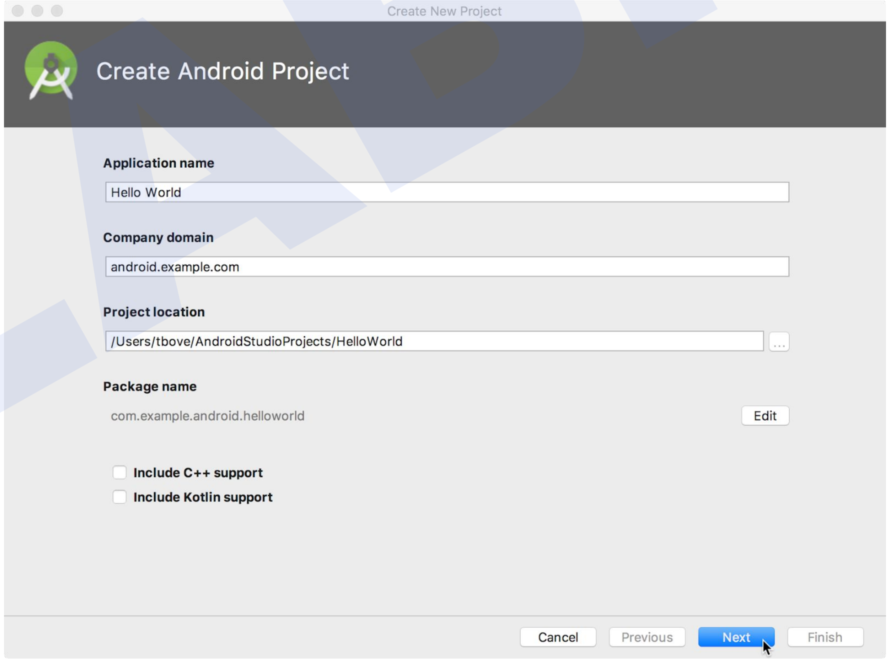
{ Google Play Publisher Account

Developer Console }

Android Application Development

Development Process - Create Project

- The company domain should be unique.
- The package name should be unique to public to google play



Create New Project

Create Android Project

Application name
Hello World

Company domain
android.example.com

Project location
/Users/tbove/AndroidStudioProjects/HelloWorld

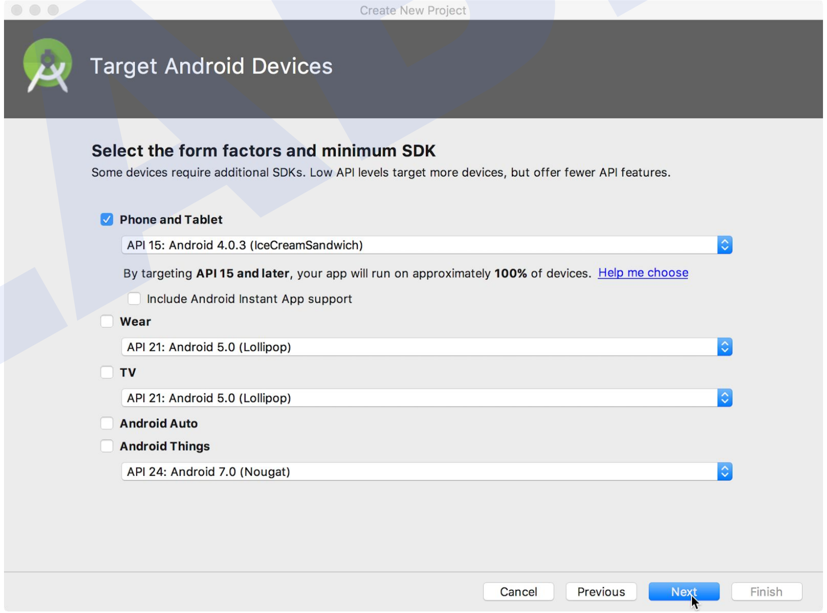
Package name
com.example.android.helloworld

□ Include C++ support
□ Include Kotlin support

Android Application Development

Development Process - Create Project

- Choose the target device



Create New Project

Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☑ **Phone and Tablet**

API 15: Android 4.0.3 (IceCreamSandwich)

By targeting **API 15 and later**, your app will run on approximately **100%** of devices. Help me choose

□ Include Android Instant App support

□ **Wear**

API 21: Android 5.0 (Lollipop)

□ **TV**

API 21: Android 5.0 (Lollipop)

□ **Android Auto**

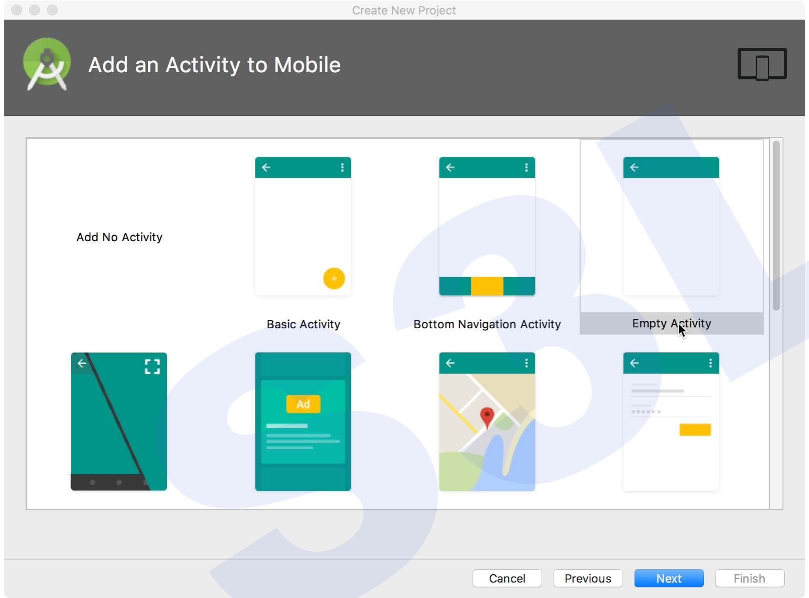
□ **Android Things**

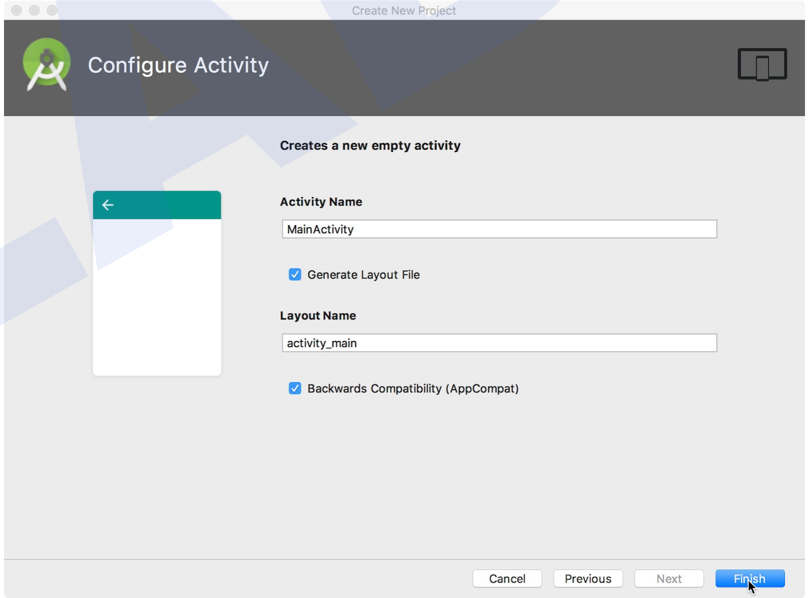
API 24: Android 7.0 (Nougat)

Android Application Development

Development Process - Create Project

- Choose the activity template





Creates a new empty activity

Activity Name

MainActivity

☑ Generate Layout File

Layout Name

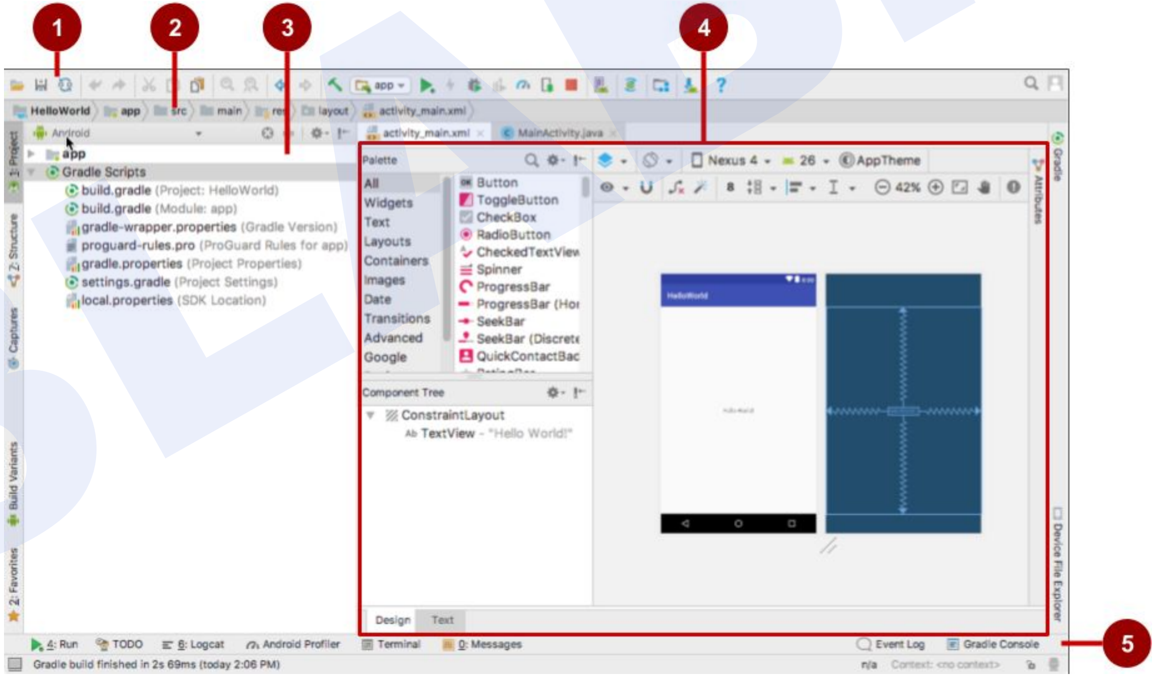
activity_main

☑ Backwards Compatibility (AppCompat)

Android Application Development

Development Process - Create Project

- Toolbar
- Navigation Bar
- Project pane
- Editor
- Tabs



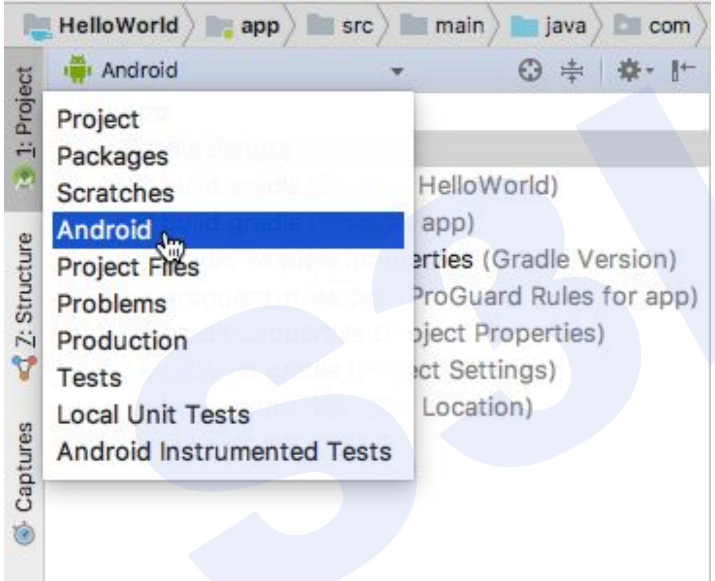
```
app
Gradle Scripts
  build.gradle (Project: HelloWorld)
  build.gradle (Module: app)
  gradle-wrapper.properties (Gradle Version)
  proguard-rules.pro (ProGuard Rules for app)
  gradle.properties (Project Properties)
  settings.gradle (Project Settings)
  local.properties (SDK Location)
```

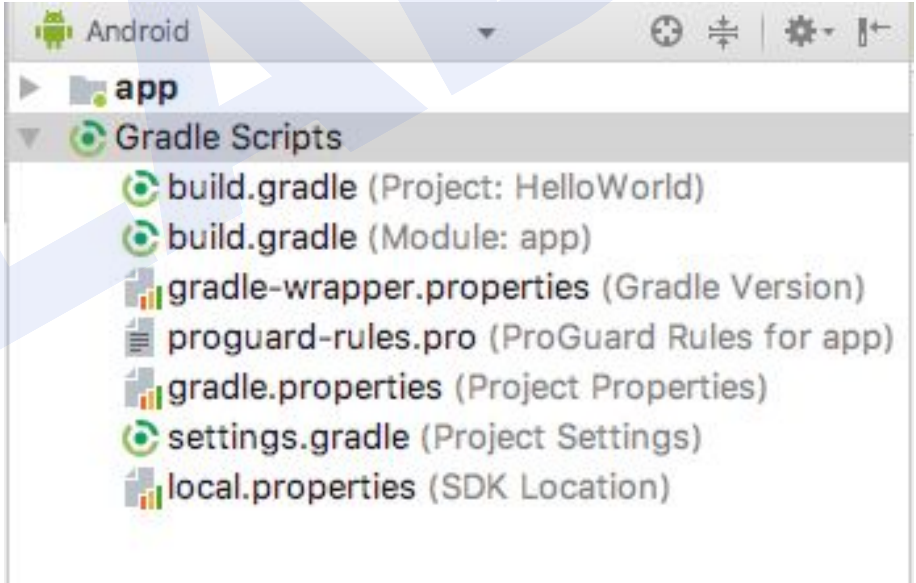
```
Gradle build finished in 2s 69ms (today 2:06 PM)
```

Android Application Development

Development Process - Create Project

- Project pane



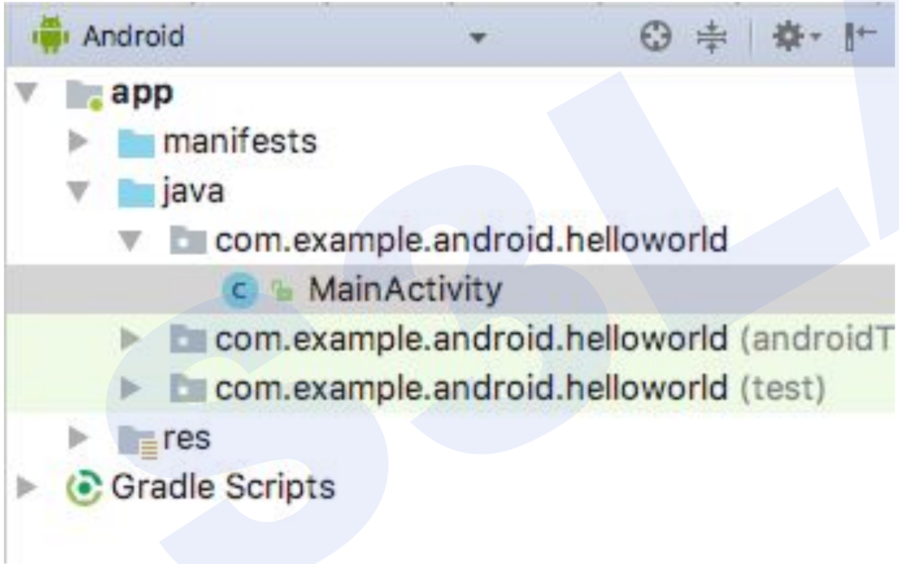


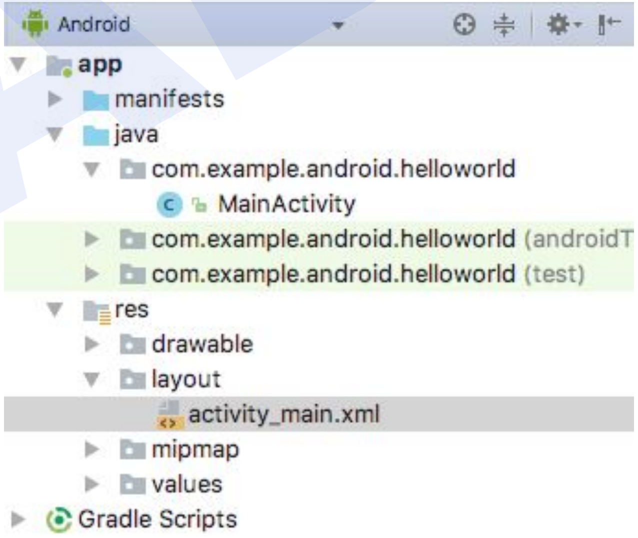
```
▶ app
▼ Gradle Scripts
    build.gradle (Project: HelloWorld)
    build.gradle (Module: app)
    gradle-wrapper.properties (Gradle Version)
    proguard-rules.pro (ProGuard Rules for app)
    gradle.properties (Project Properties)
    settings.gradle (Project Settings)
    local.properties (SDK Location)
```

Android Application Development

Development Process - Create Project

- Project pane

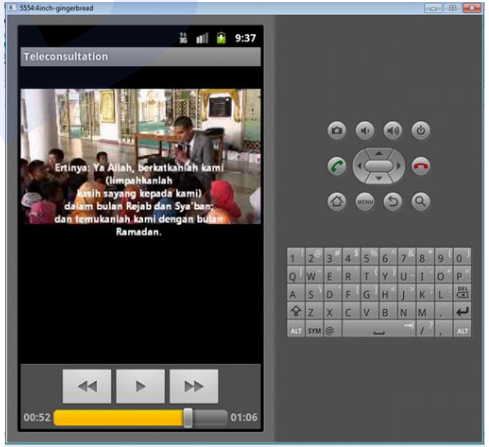

Android
app
manifests
java
com.example.android.helloworld
MainActivity
com.example.android.helloworld (androidT
com.example.android.helloworld (test)
res
Gradle Scripts


Android
app
manifests
java
com.example.android.helloworld
MainActivity
com.example.android.helloworld (androidT
com.example.android.helloworld (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts

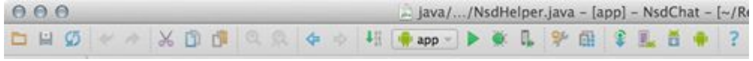
Android Application Development

Android Studio - Emulator

- Two choices for deployment:
    - Real Android device
    - Android virtual device
- Plug in your real device; otherwise, create an Android virtual device
- Emulator is slow. Try Intel accelerated version, or perhaps http://www.genymotion.com/
- Run the app: press “Run” button in toolbar



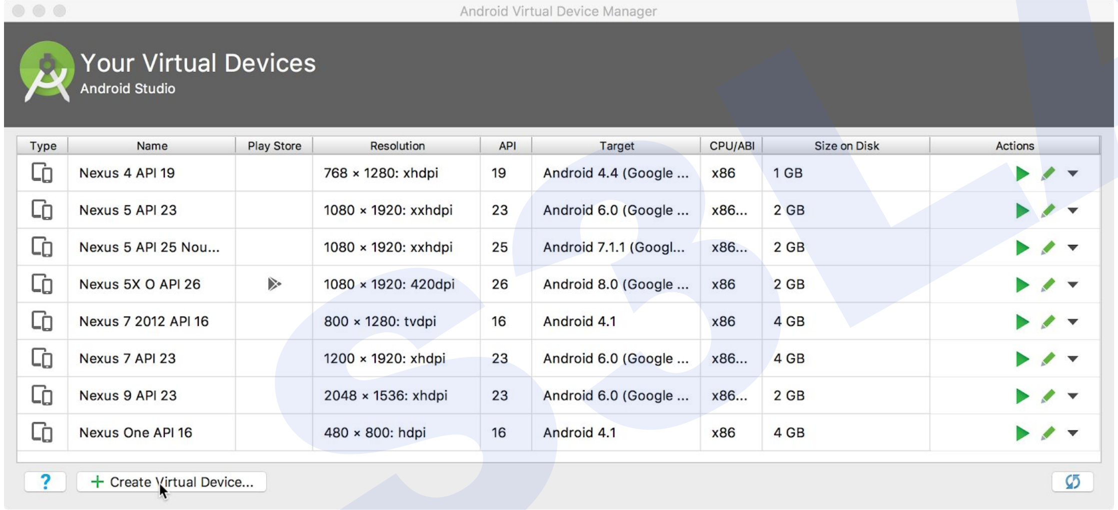
Ertinya: Ya Allah, berkatkanlah kami (limpahkanlah kasih sayang kepada kami) dalam bulan Rejab dan Sya'ban; dan temukanlah kami dengan bulan Ramadan.



Android Application Development

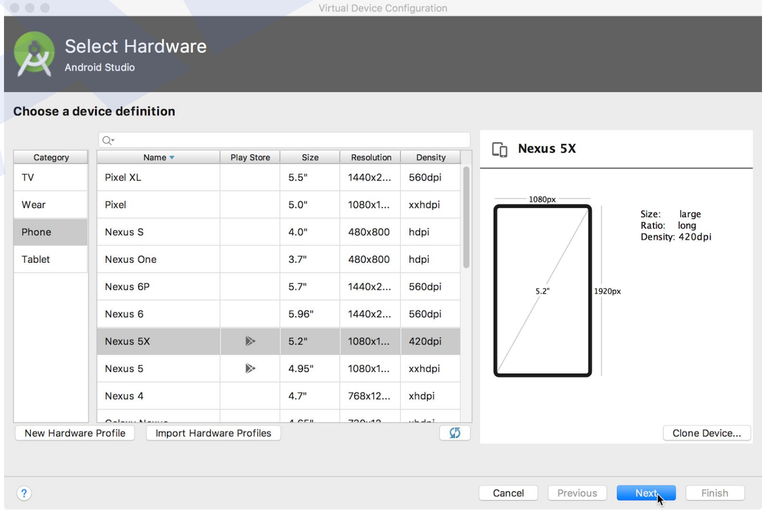
Android Studio - Emulator

- Tools > Android > AVD Manager



Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 4 API 19		768 × 1280: xhdpi	19	Android 4.4 (Google ...	x86	1 GB	
	Nexus 5 API 23		1080 × 1920: xxhdpi	23	Android 6.0 (Google ...	x86...	2 GB	
	Nexus 5 API 25 Nou...		1080 × 1920: xxhdpi	25	Android 7.1.1 (Googl...	x86...	2 GB	
	Nexus 5X O API 26	▶	1080 × 1920: 420dpi	26	Android 8.0 (Google ...	x86	2 GB	
	Nexus 7 2012 API 16		800 × 1280: tvdpi	16	Android 4.1	x86	4 GB	
	Nexus 7 API 23		1200 × 1920: xhdpi	23	Android 6.0 (Google ...	x86...	4 GB	
	Nexus 9 API 23		2048 × 1536: xhdpi	23	Android 6.0 (Google ...	x86...	2 GB	
	Nexus One API 16		480 × 800: hdpi	16	Android 4.1	x86	4 GB	



Select Hardware
Android Studio

Choose a device definition

Category
TV
Wear
Phone
Tablet

Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2...	560dpi
Pixel		5.0"	1080x1...	xxhdpi
Nexus S		4.0"	480x800	hdpi
Nexus One		3.7"	480x800	hdpi
Nexus 6P		5.7"	1440x2...	560dpi
Nexus 6		5.96"	1440x2...	560dpi
Nexus 5X	▶	5.2"	1080x1...	420dpi
Nexus 5	▶	4.95"	1080x1...	xxhdpi
Nexus 4		4.7"	768x12...	xhdpi

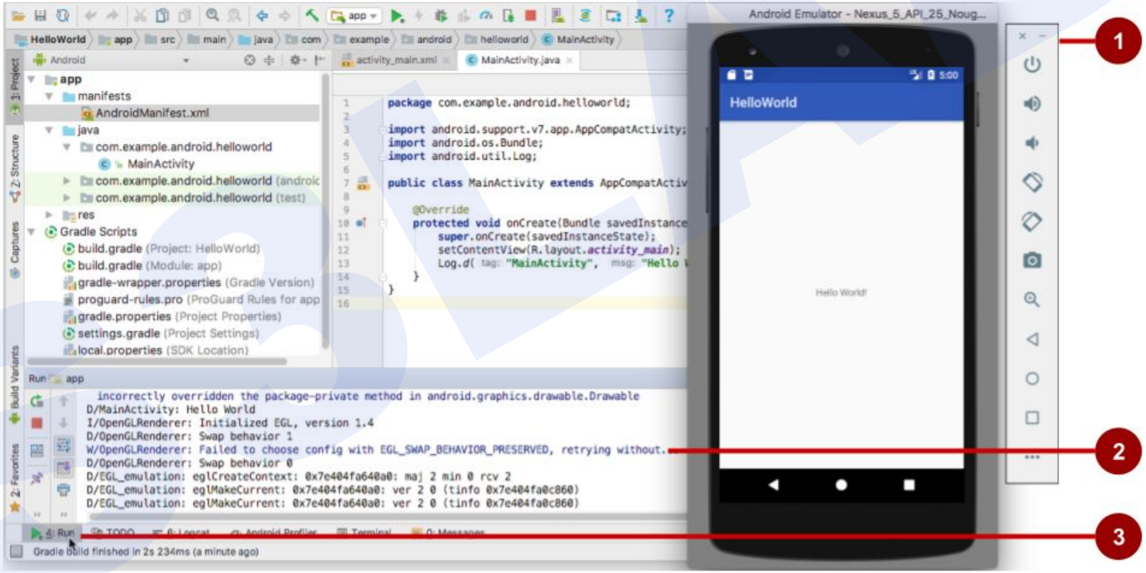
Nexus 5X

Size: large
Ratio: long
Density: 420dpi

Android Application Development

Android Studio - Emulator

- Tools > Android > AVD Manager



```java
package com.example.android.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActiv

    @Override
    protected void onCreate(Bundle savedInstance
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d( tag: "MainActivity",  msg: "Hello W
    }
}
```

```
  incorrectly overridden the package-private method in android.graphics.drawable.Drawable
D/MainActivity: Hello World
I/OpenGLRenderer: Initialized EGL, version 1.4
D/OpenGLRenderer: Swap behavior 1
W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without.
D/OpenGLRenderer: Swap behavior 0
D/EGL_emulation: eglCreateContext: 0x7e404fa640a0: maj 2 min 0 rcv 2
D/EGL_emulation: eglMakeCurrent: 0x7e404fa640a0: ver 2 0 (tinfo 0x7e404fa0c860)
D/EGL_emulation: eglMakeCurrent: 0x7e404fa640a0: ver 2 0 (tinfo 0x7e404fa0c860)
```

Gradle build finished in 2s 234ms (a minute ago)

Android Emulator - Nexus_5_API_25_Noug...
HelloWorld
app
src
main
java
com
example
android
helloworld
MainActivity
Android
activity_main.xml
MainActivity.java
app
manifests
AndroidManifest.xml
java
com.example.android.helloworld
MainActivity
com.example.android.helloworld (androidTest)
com.example.android.helloworld (test)
res
Gradle Scripts
build.gradle (Project: HelloWorld)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
2: Structure
Captures
Build Variants
2: Favorites
Run
app
4: Run
TODO
6: Logcat
Android Profiler
Terminal
0: Messages
5:00
HelloWorld
Hello World!
1
2
3

Android Application Development

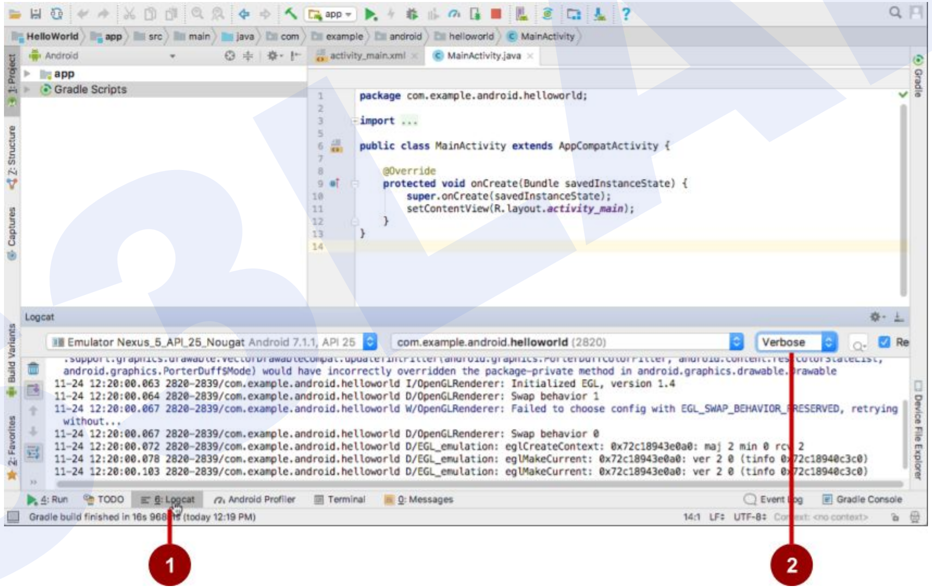
Android Studio - Actual Device Debugging

- USB data cable
- Android USB driver – need to install on PC
- Enable Developers Options
  - Setting -> About Phone -> 7x taps on Build Number
- Enable USB debugging
  - Settings -> Developer options -> Enable USB Debugging

Android Application Development

Android Studio - Logcat

- Log.d(“filter message”, “message content”);



```
package com.example.android.helloworld;

import ...

public class MainActivity extends AppCompatActivity {

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

Emulator Nexus_5_API_25_Nougat Android 7.1.1, API 25 | com.example.android.helloworld (2820) | Verbose | ☑ Re

```
 android.graphics.PorterDuff$Mode) would have incorrectly overridden the package-private method in android.graphics.drawable.Drawable
11-24 12:20:00.063 2820-2839/com.example.android.helloworld I/OpenGLRenderer: Initialized EGL, version 1.4
11-24 12:20:00.064 2820-2839/com.example.android.helloworld D/OpenGLRenderer: Swap behavior 1
11-24 12:20:00.067 2820-2839/com.example.android.helloworld W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying
 without...
11-24 12:20:00.067 2820-2839/com.example.android.helloworld D/OpenGLRenderer: Swap behavior 0
11-24 12:20:00.072 2820-2839/com.example.android.helloworld D/EGL_emulation: eglCreateContext: 0x72c18943e0a0: maj 2 min 0 rcv 2
11-24 12:20:00.078 2820-2839/com.example.android.helloworld D/EGL_emulation: eglMakeCurrent: 0x72c18943e0a0: ver 2 0 (tinfo 0x72c18940c3c0)
11-24 12:20:00.103 2820-2839/com.example.android.helloworld D/EGL_emulation: eglMakeCurrent: 0x72c18943e0a0: ver 2 0 (tinfo 0x72c18940c3c0)
```

Gradle build finished in 16s 968ms (today 12:19 PM)   14:1 LF÷ UTF-8÷ Context: <no context>

Homework

- Install **Android Studio**: https://developer.android.com/studio
- Create the First Android Application, Run it on emulator and Real-device.

  https://developer.android.com/codelabs/build-your-first-android-app#2

Q & A

Thank you for listening

“Coming together is a beginning;
Keeping together is progress;
Working together is success.”
- HENRY FORD