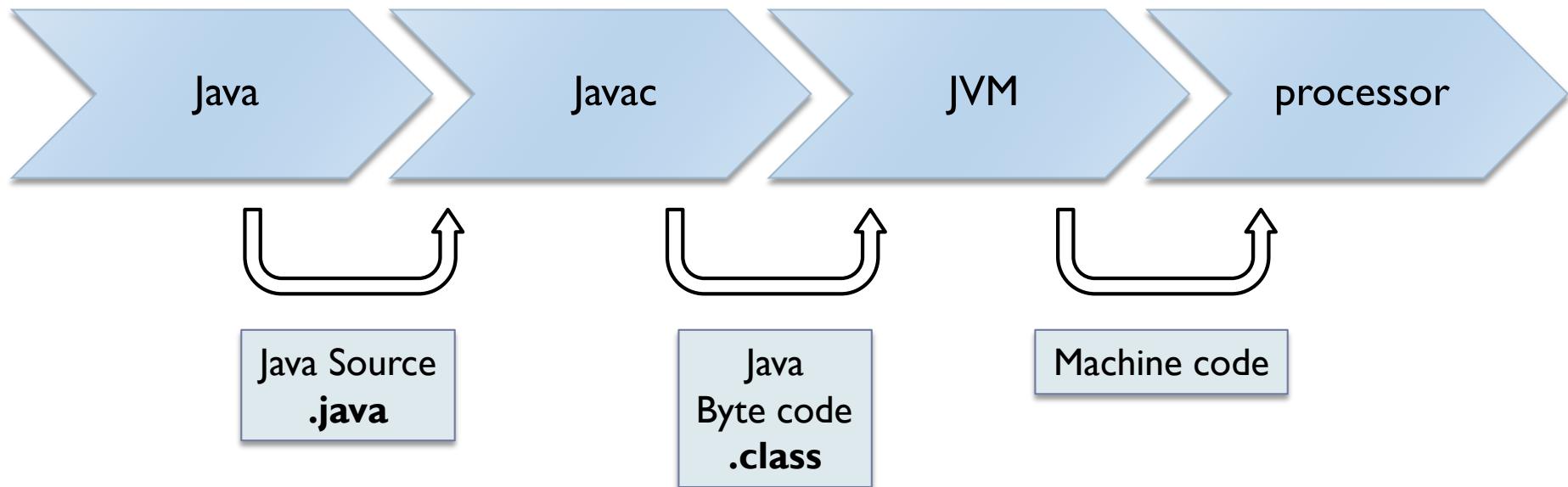


# Android Application Development

Amit Gulati, amit.gulati@gmail.com

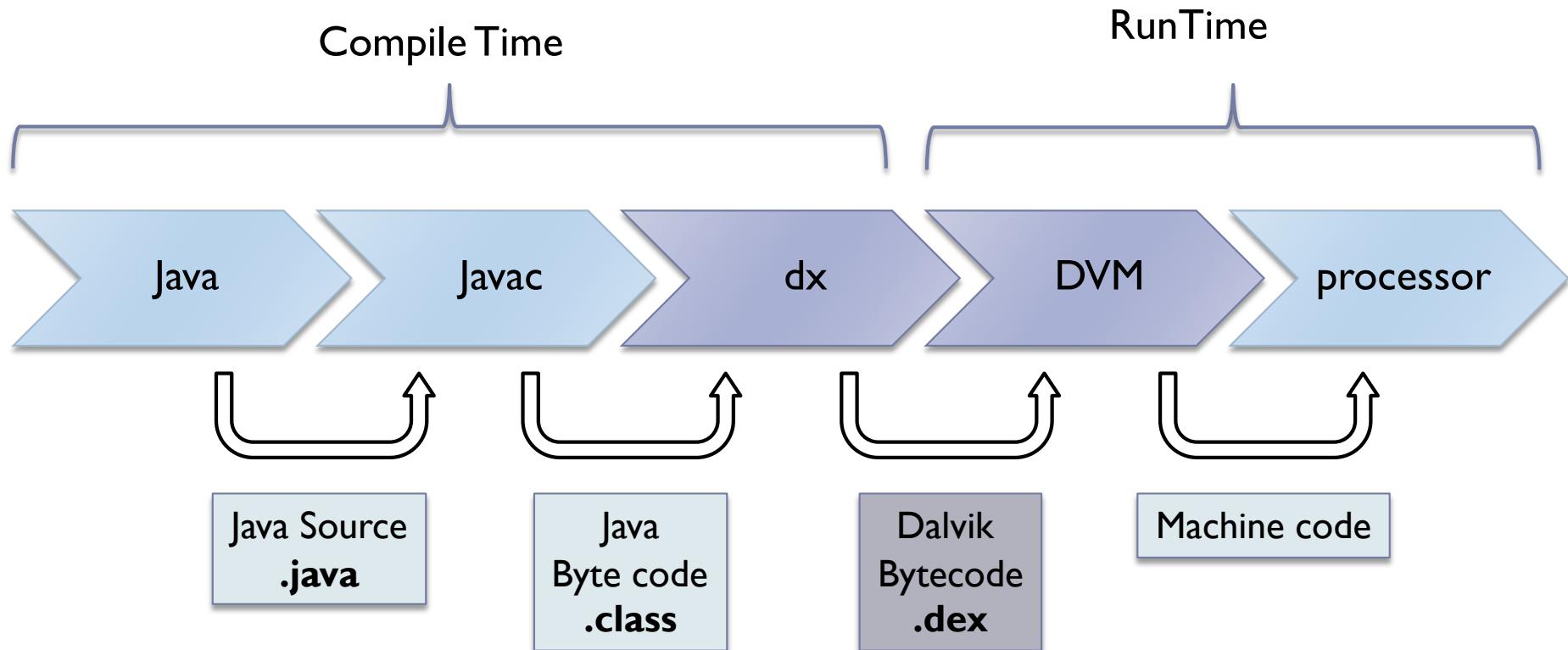
# Android Architecture

- ▶ Java Virtual Machine
  - ▶ Java byte code requires an JVM that converts the byte code to native instructions



# Android Architecture

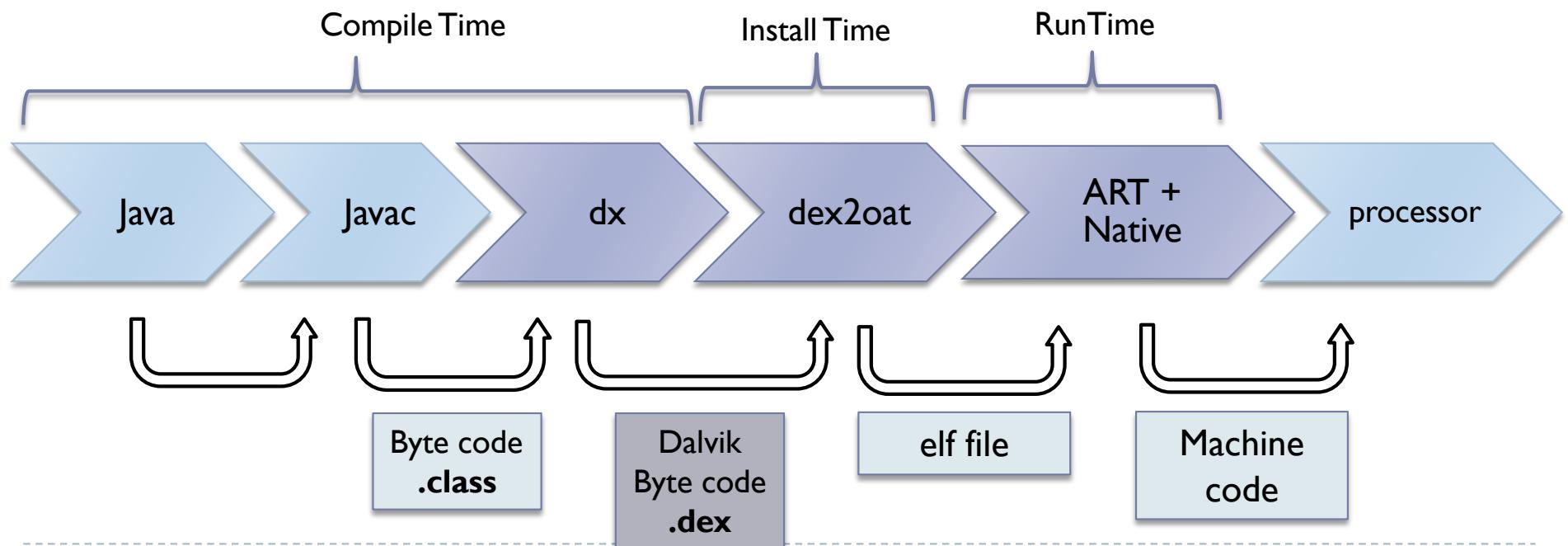
- ▶ **Dalvik Virtual Machine**
  - ▶ Default with Kitkat and below
  - ▶ Optimized for mobile and embedded devices
  - ▶ Runs Dalvik byte code or “**.dex**” files.



# Android Architecture

## ▶ ART

- ▶ Default with Lollipop and above
- ▶ Ahead-of-time (AOT)
- ▶ Improved Garbage Collection
- ▶ 64bit support





# Android Studio



Amit Gulati, amit.gulati@gmail.com

# Android Studio

---

- ▶ Installation
  - ▶ Download location

<https://developer.android.com/studio>

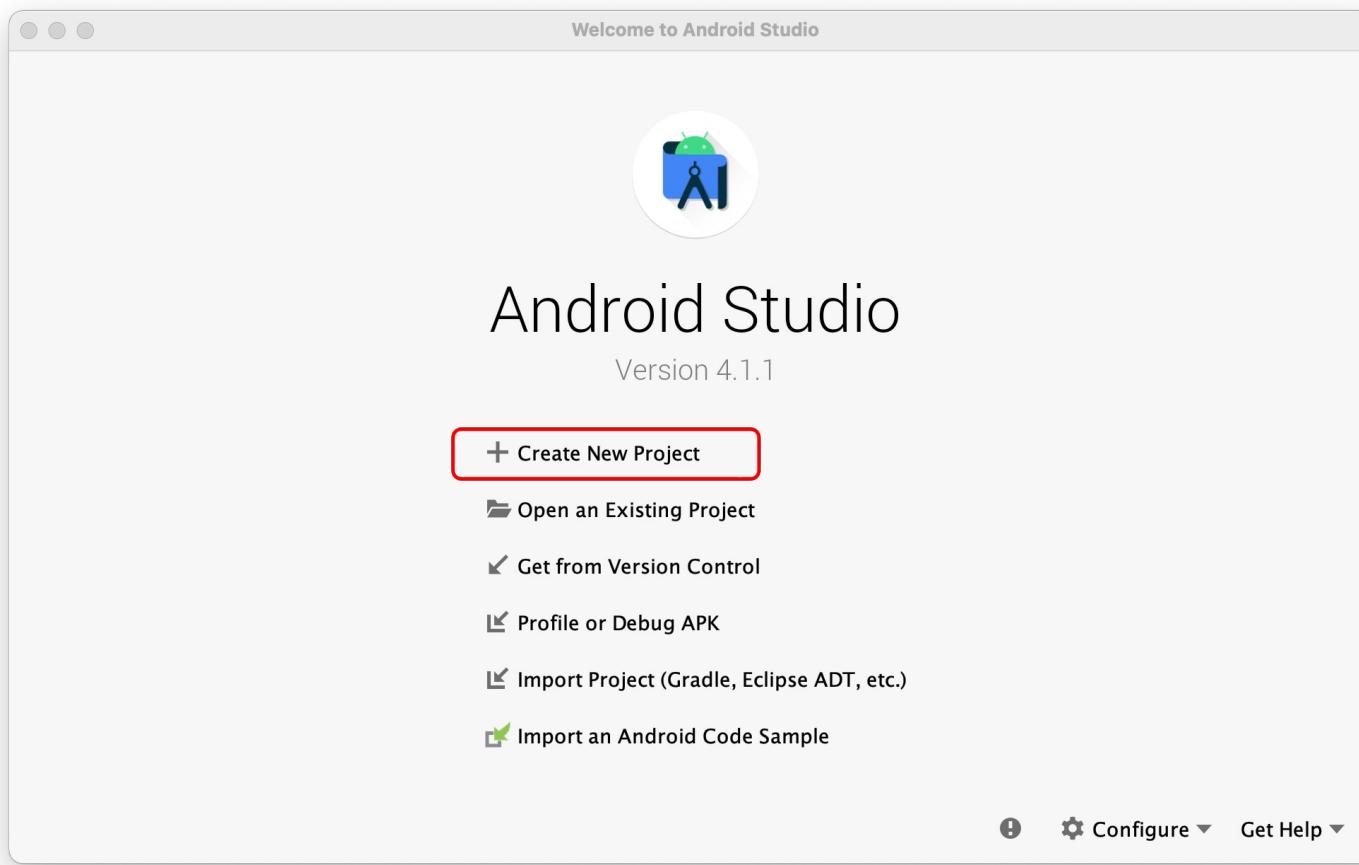


Android Studio provides the fastest tools for building apps on every type of Android device.

[DOWNLOAD ANDROID STUDIO](#)

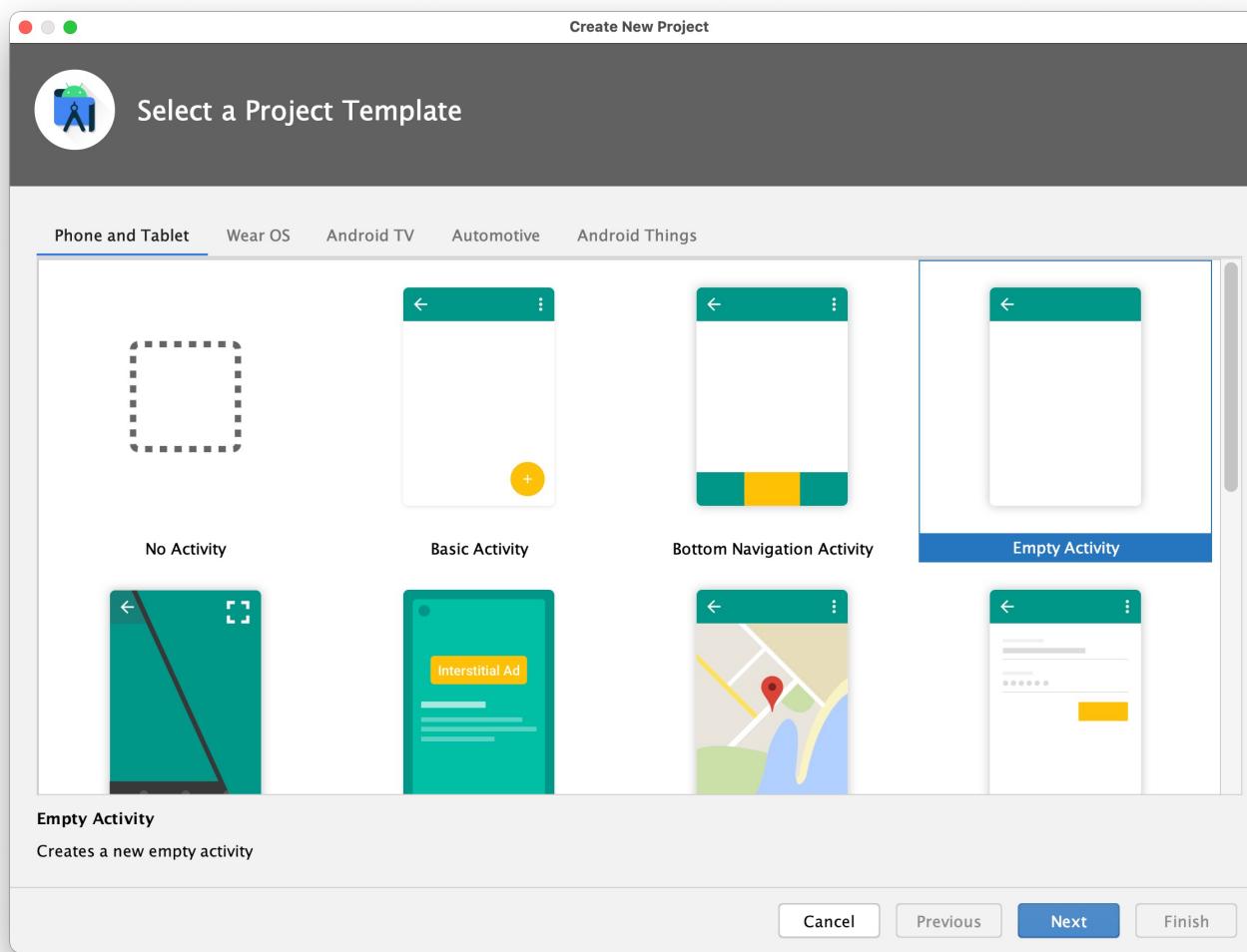
# Android Studio

- ▶ Start Android Studio
  - ▶ Select “Create New Project”



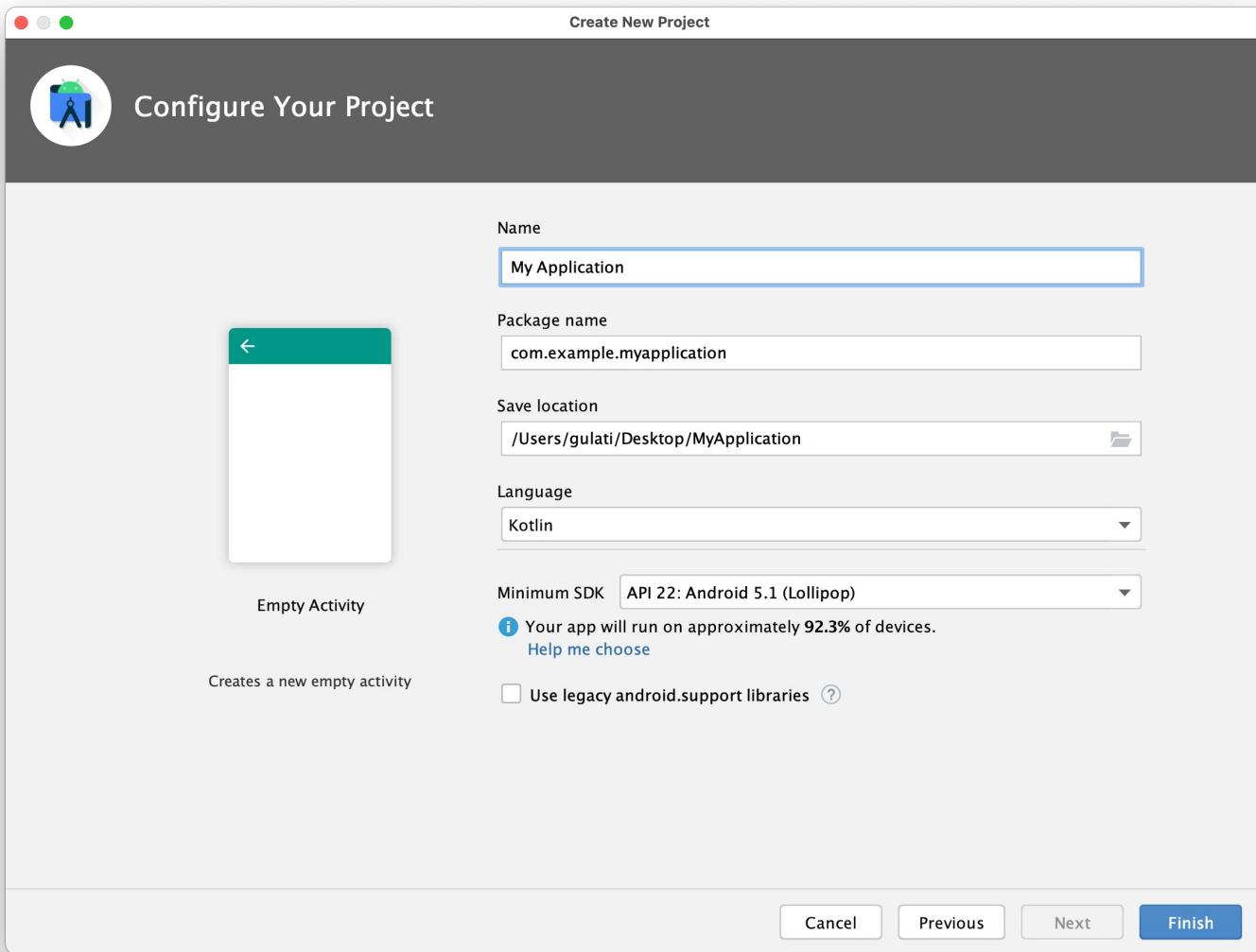
# Android Studio

## ▶ Project Template

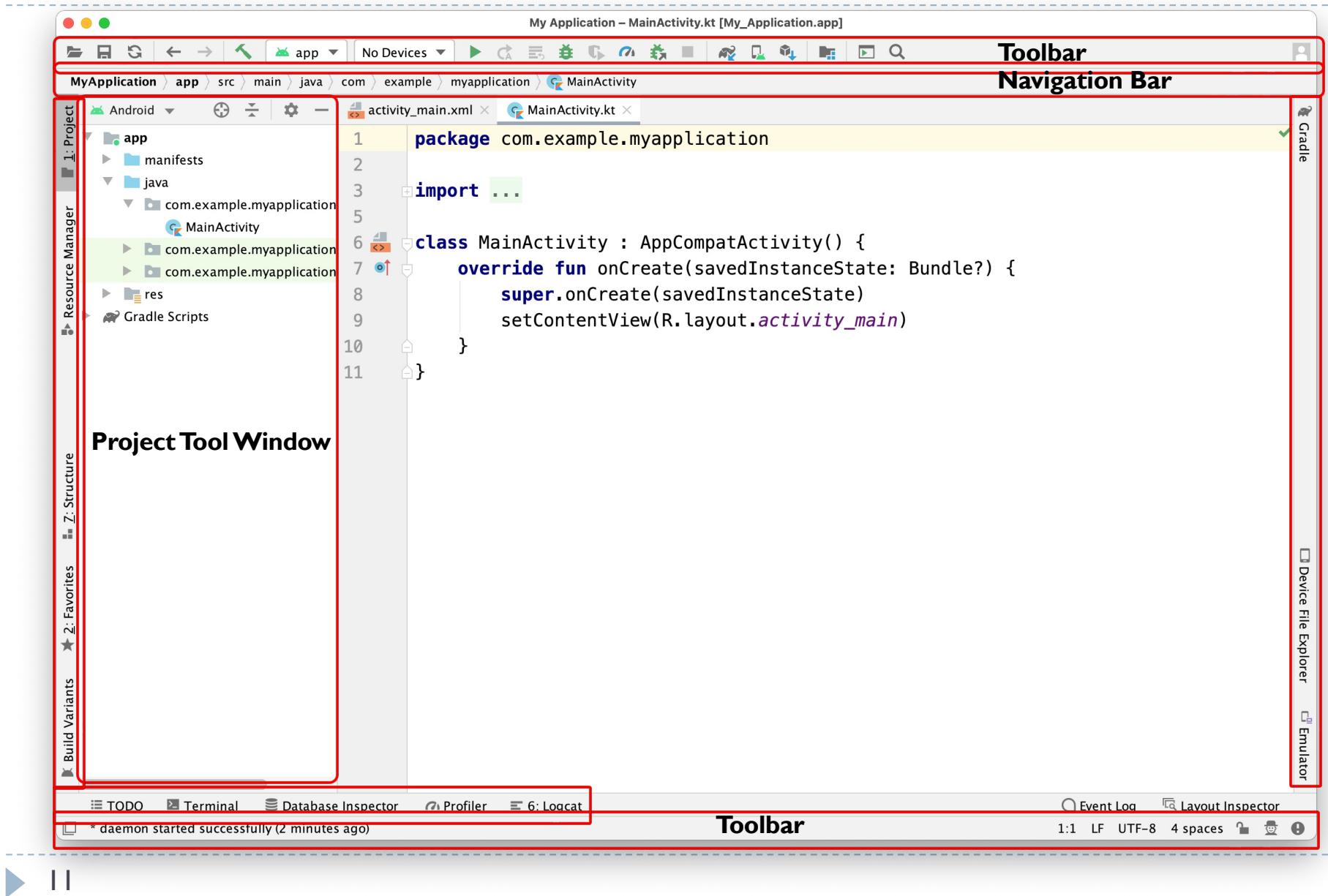


# Android Studio

## ▶ Project Settings



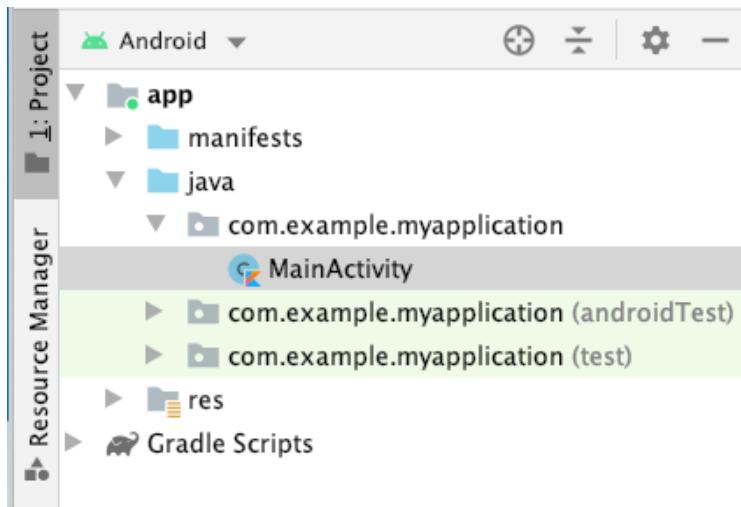
# Android Studio



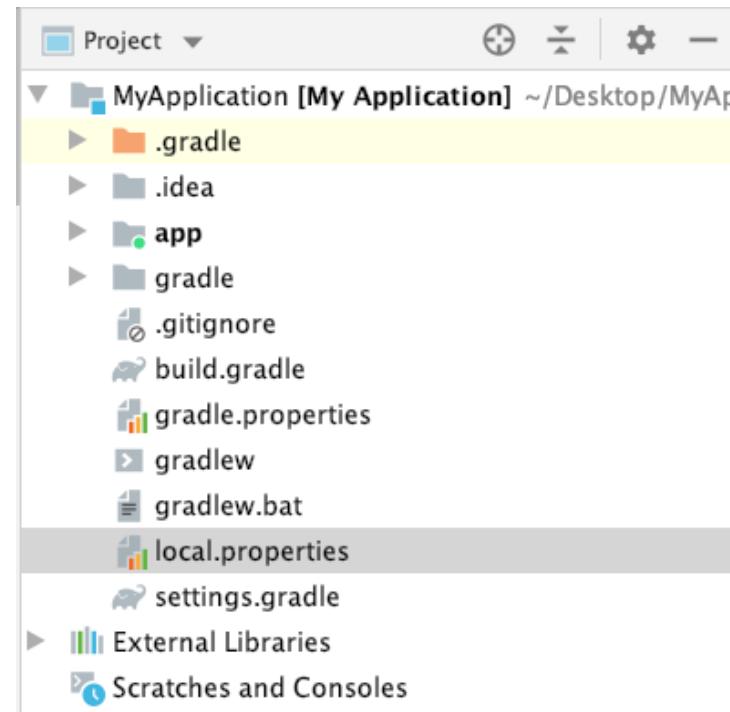
# Android Studio

## ▶ Project Tool Window

Android View

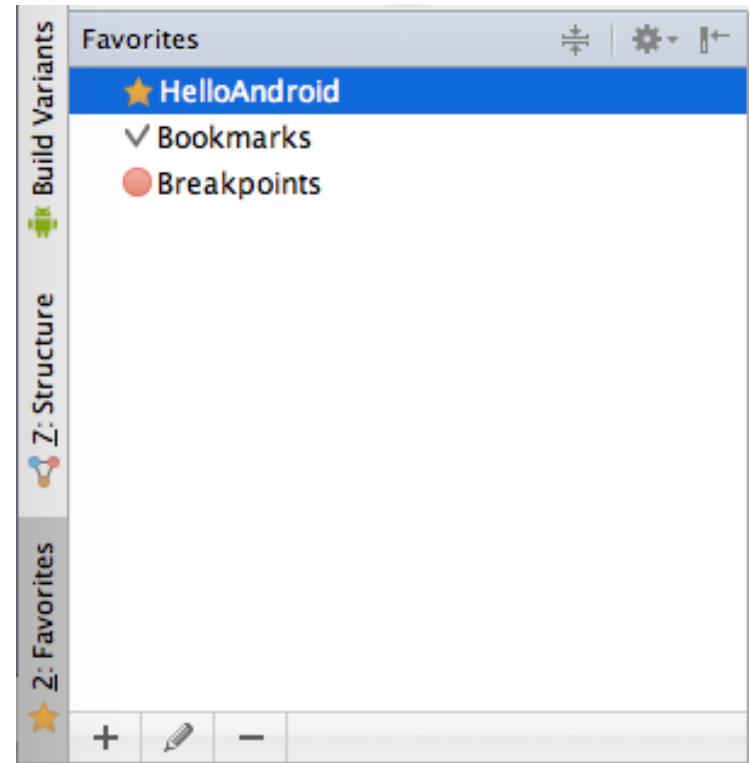


Project View



# Android Studio

- ▶ Favorite Tool Window
  - ▶ Logically group references to related files
  - ▶ Contains 3 sections
    - ▶ Favorites
      - Files can be added to Favorites sections.
    - ▶ Bookmarks
      - Allow you to navigate to any line in a file.
    - ▶ Breakpoints

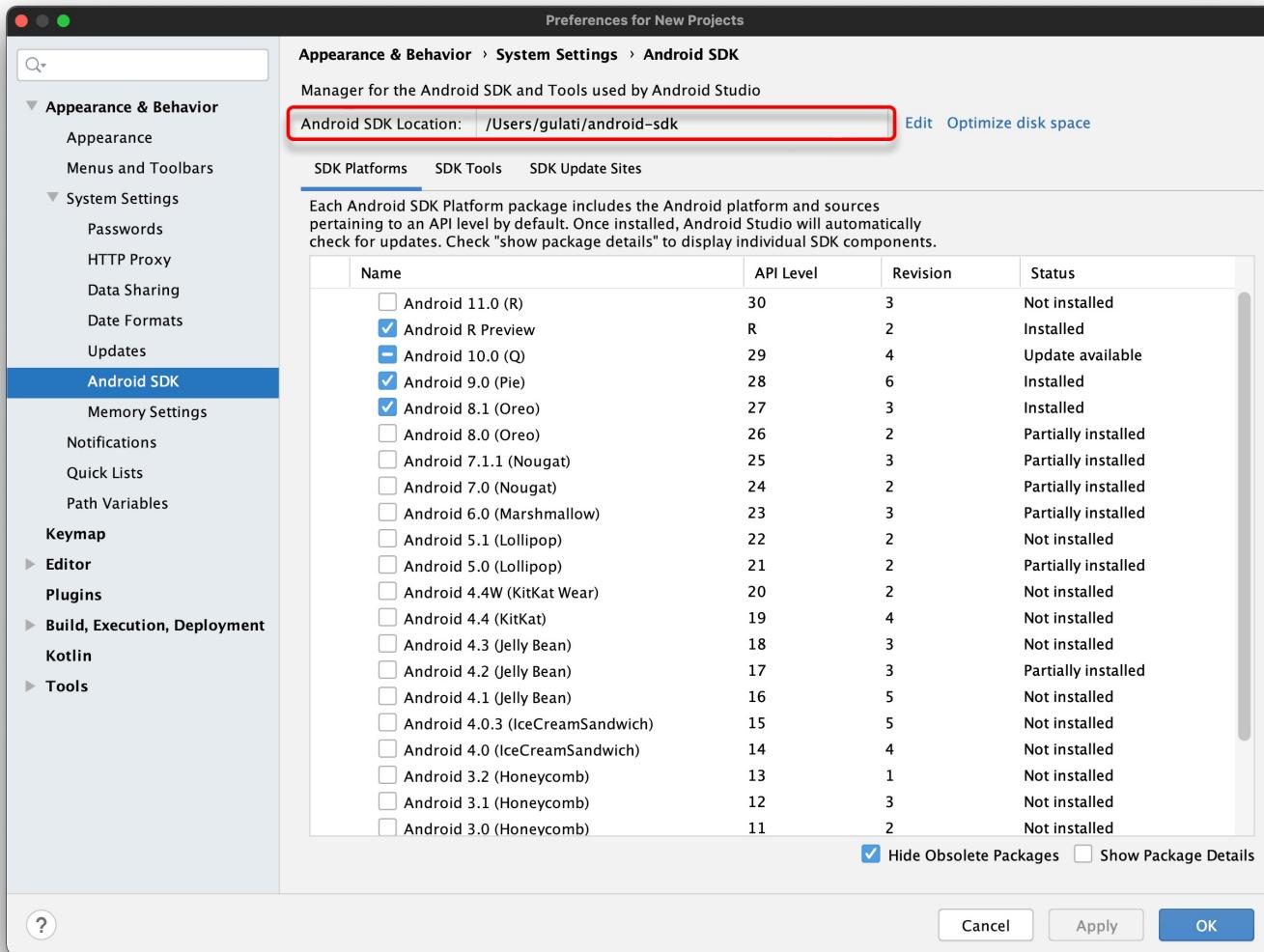


Favorite ( ⇧↑ f )

Bookmark (fn F3)

# Android Studio

## ▶ SDK Manager

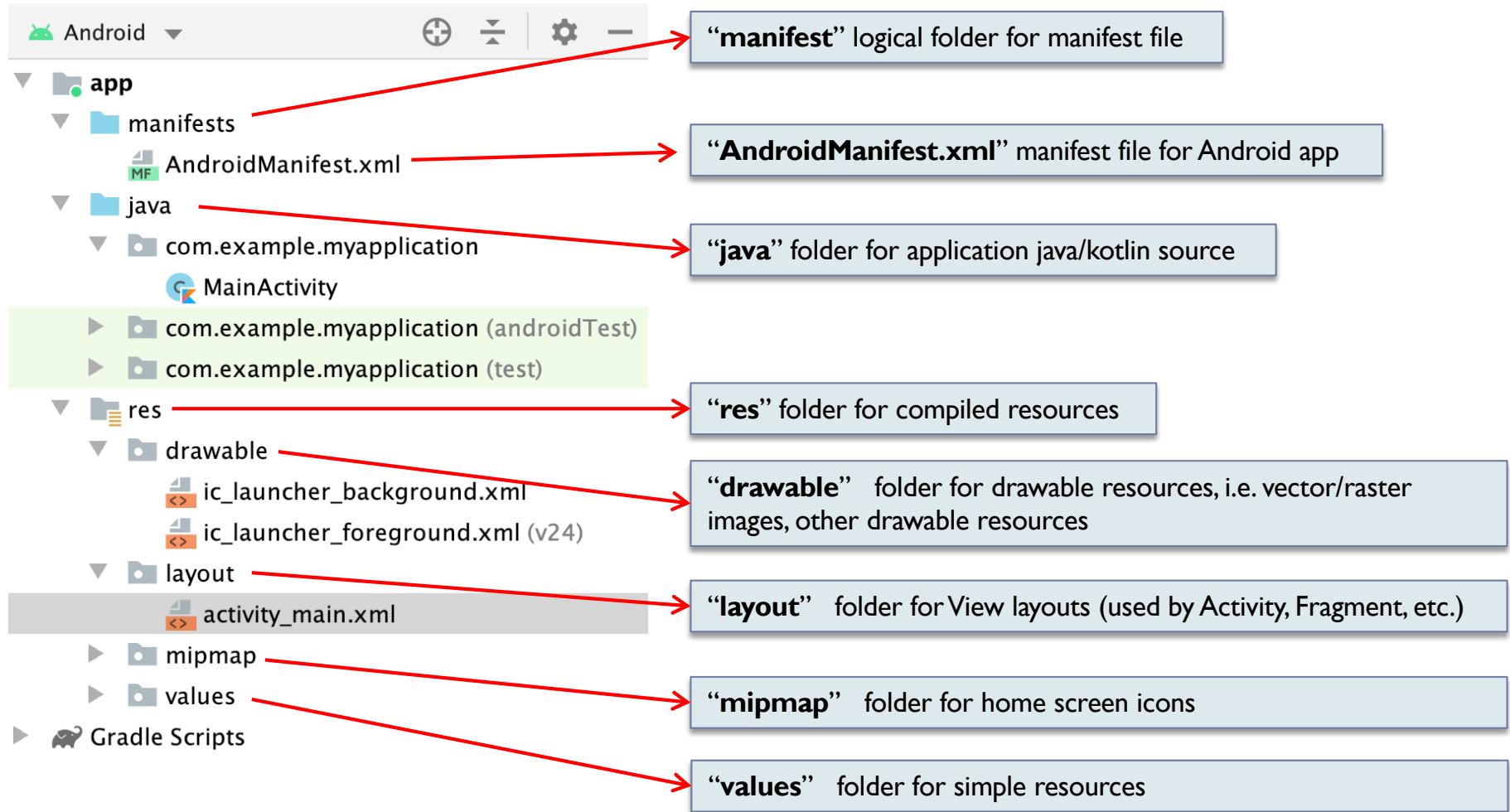


# Android Project

amit.gulati@gmail.com

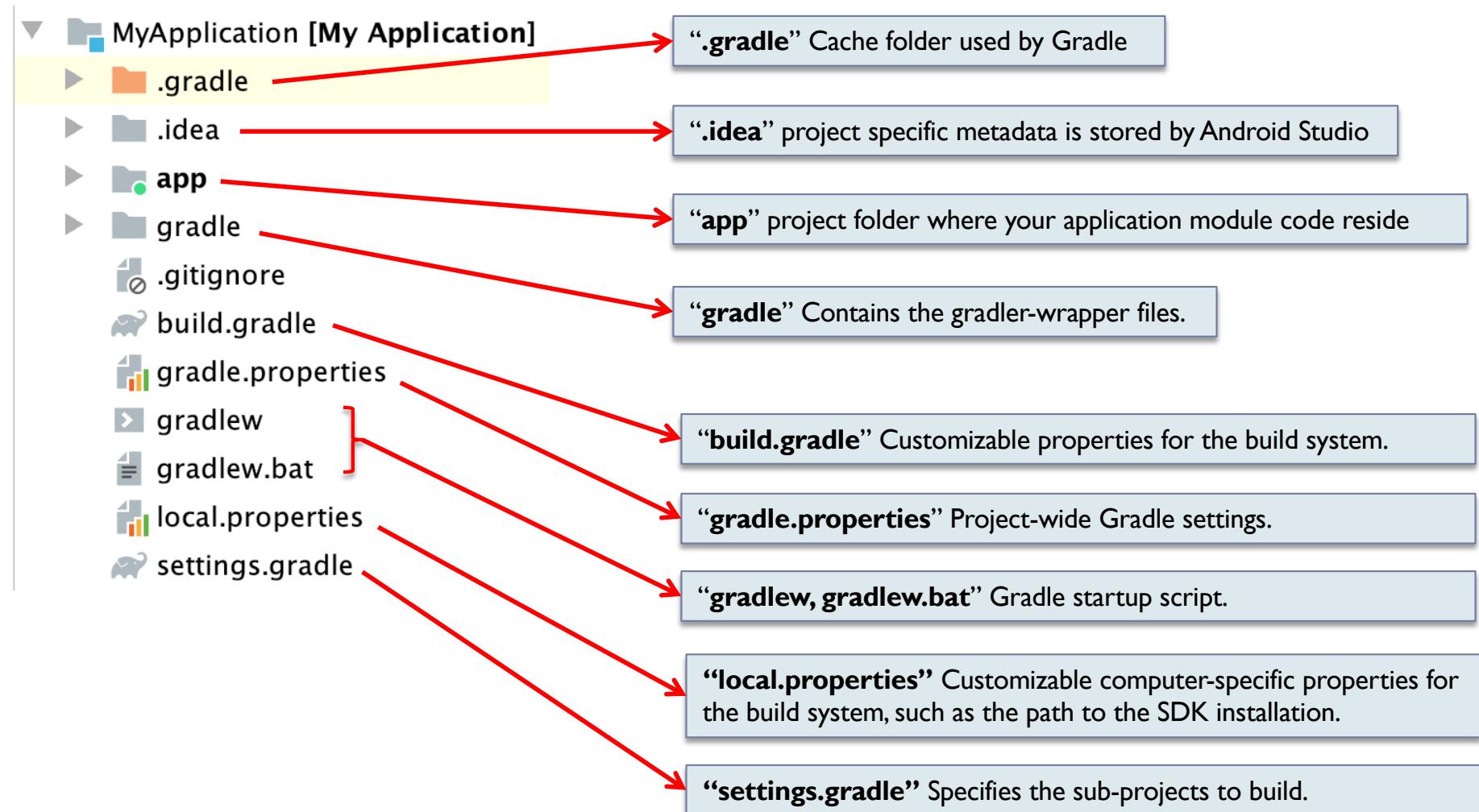
# Android Studio

## ▶ Android View



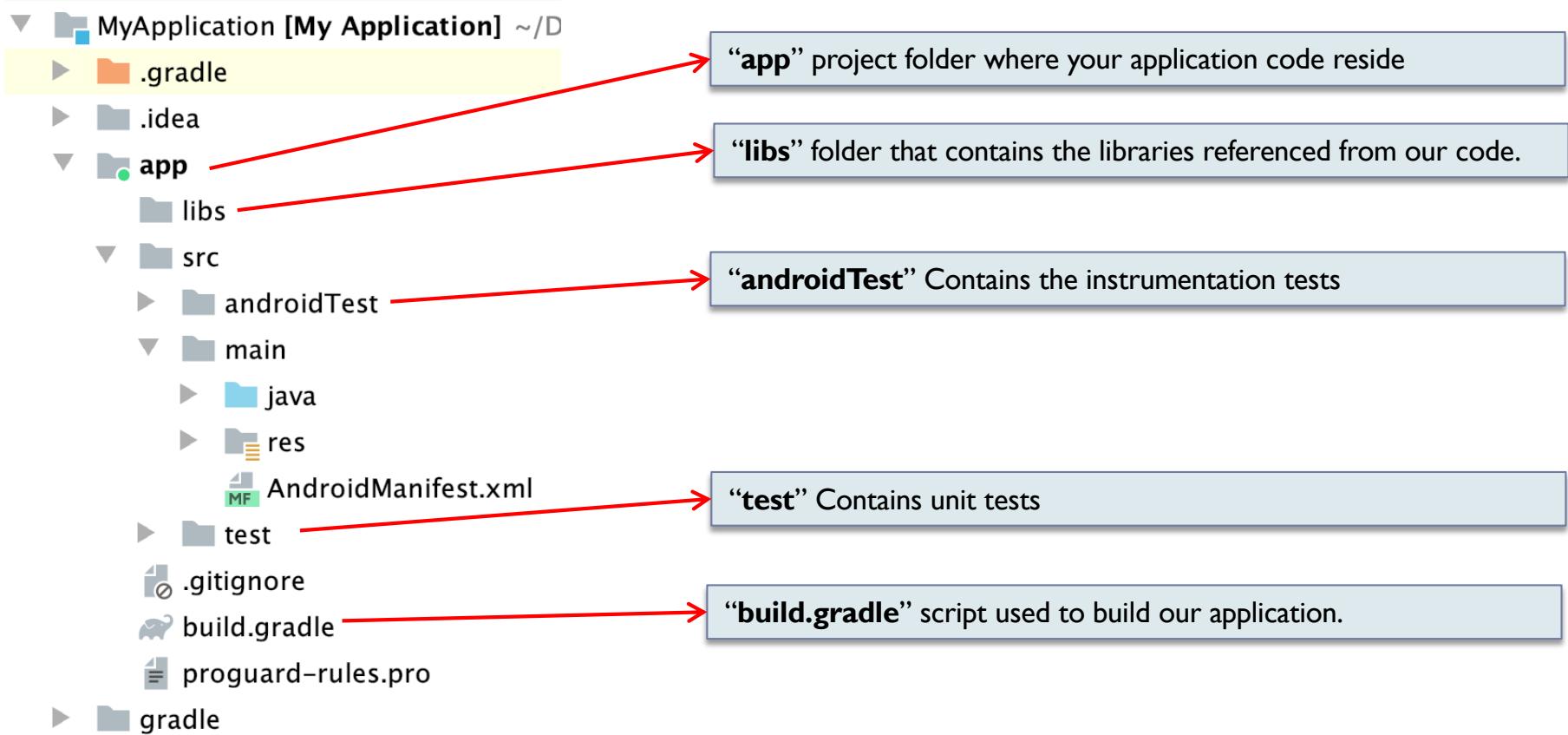
# Android Studio

## ▶ Project View: File/Folder structure



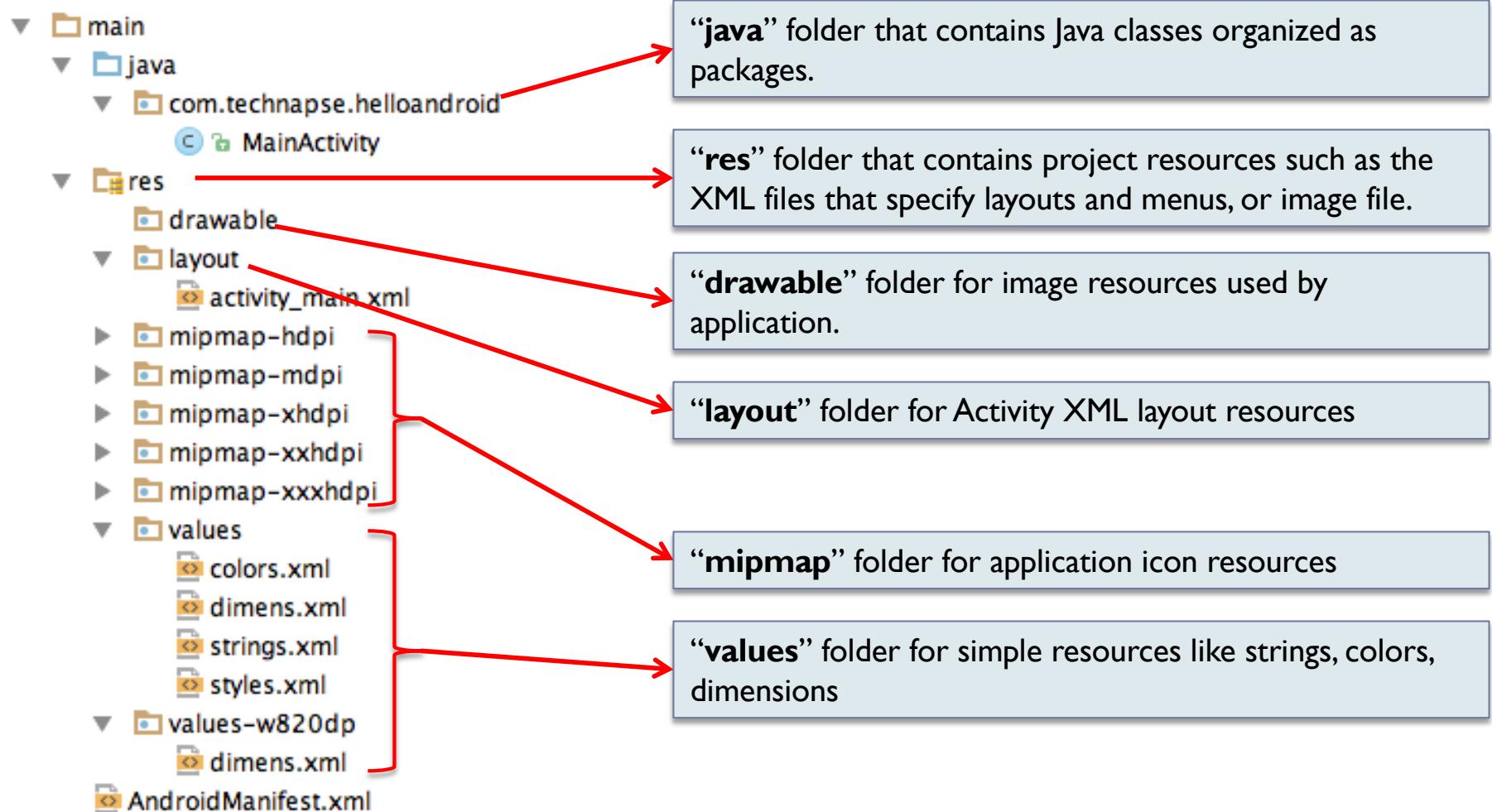
# Android Studio

## ▶ Project View: File/Folder structure



# Android Studio

## ▶ Project View: File/Folder structure





# Application Essentials

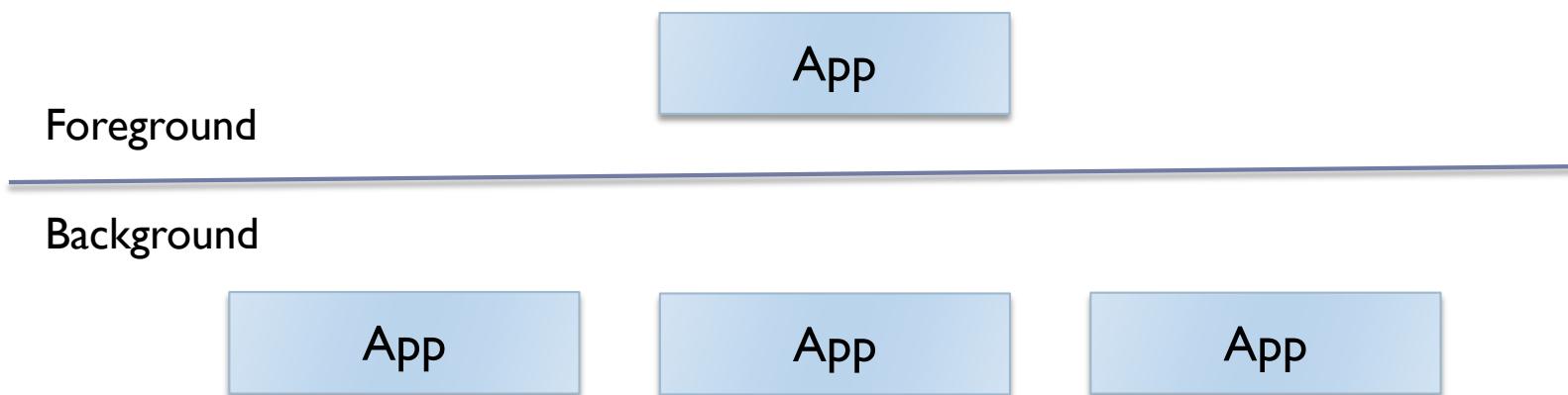


Amit Gulati, amit.gulati@gmail.com

# Android Runtime

---

- ▶ Active vs Background Applications
  - ▶ Active application
    - ▶ Application that is currently displaying content on the screen and has focus
  - ▶ Background application
    - ▶ Application process is still in Memory but is not displaying content on the screen.
    - ▶ Application has restricted access to CPU



# Application Essentials

## ▶ Physical Structure

- ▶ An Android application is bundled as a “.apk” file.
- ▶ .apk if a zip file



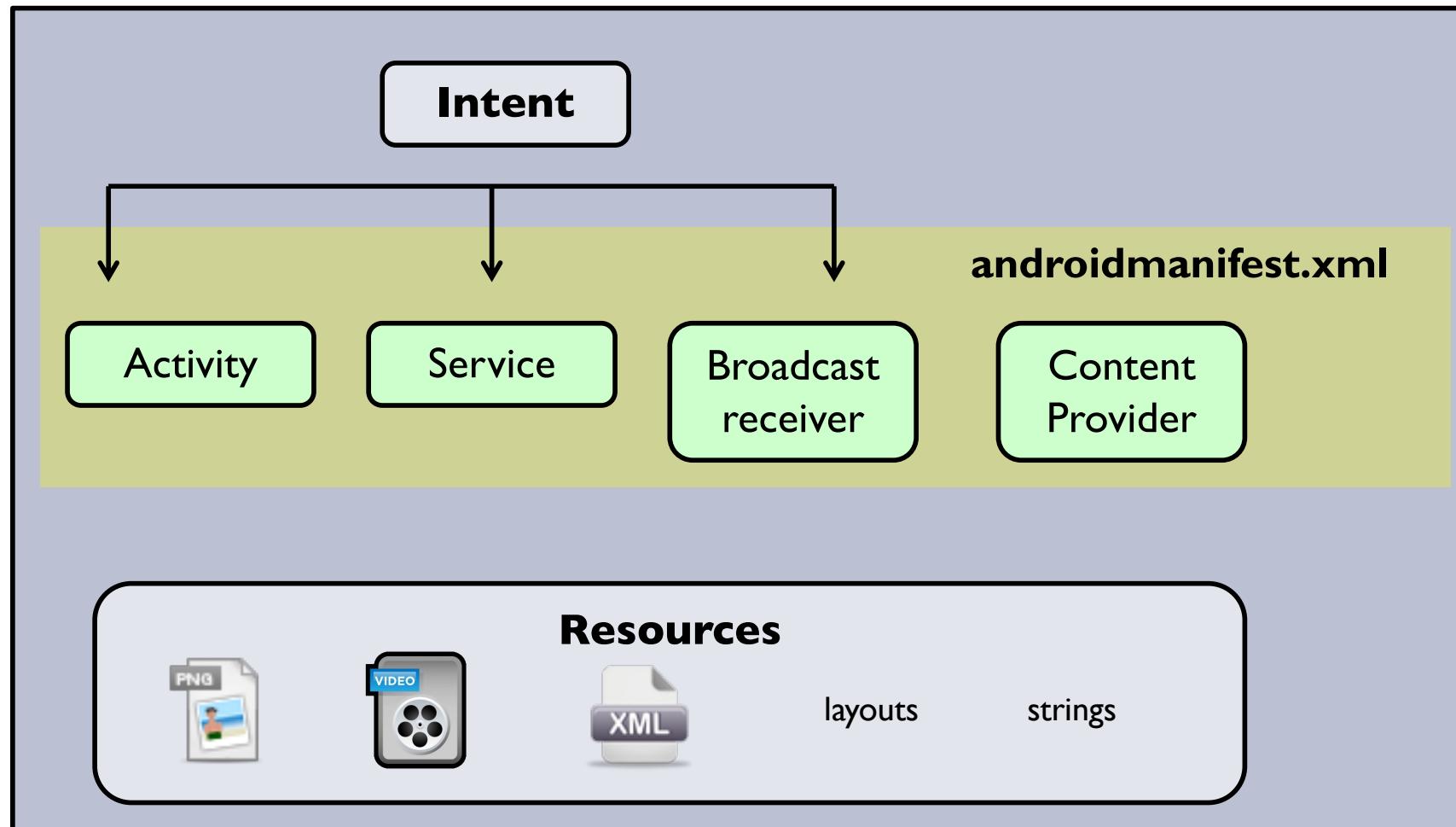
# Application Essentials

---

- ▶ What is an Android Application ?
  - ▶ Collection of loosely coupled Application Components
    - ▶ Activity
    - ▶ Service
    - ▶ Content Provider
    - ▶ Broadcast Receiver
  - ▶ Bound together by the Application Manifest.
  - ▶ Communicate with each other via Intent objects.
  - ▶ Use Resources
    - ▶ Almost everything other than Java code in Android is a resource.
    - ▶ Common Resources: String, Images, Audio, Video, Layouts etc.

# Application Essentials

- ▶ What is an Android Application ?

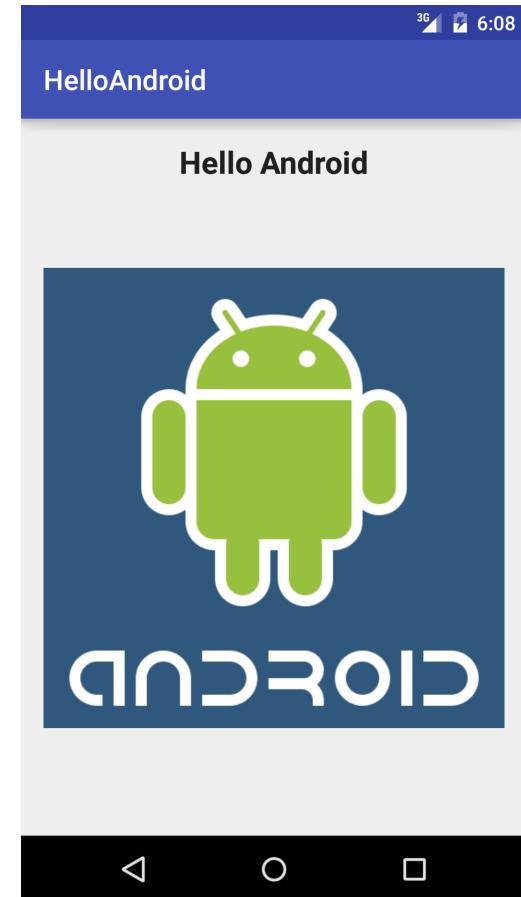


# Application Essentials

## ▶ Activity

- ▶ Activities form the Presentation layer of an Android application.
- ▶ A Single Screen in your application is an Activity.
  - ▶ Covers the whole screen of device
- ▶ Activity object is given a blank window to draw.
  - ▶ Activity object is not capable of drawing in a window.

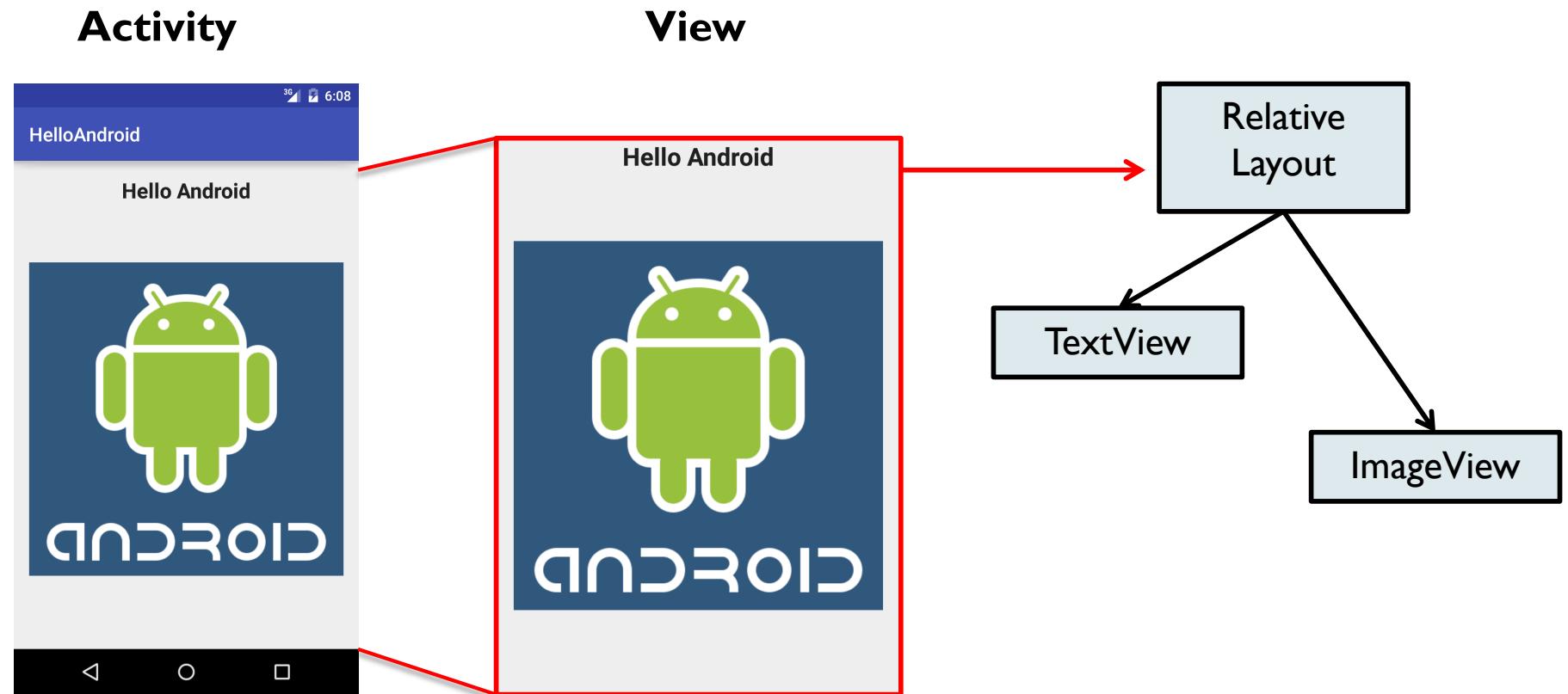
## Simple Activity



# Application Essentials

## ▶ Activity

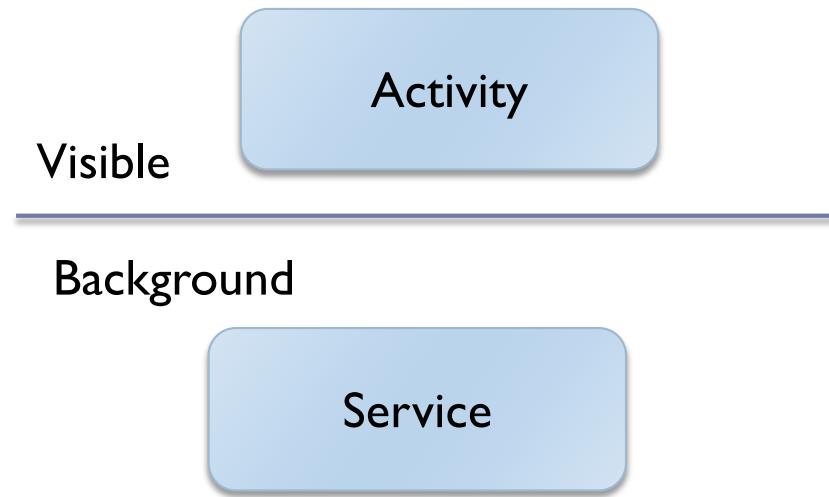
- ▶ Activity attaches a **View** object to itself
  - ▶ A View object can draw in the window.



# Application Essentials

---

- ▶ **Service**
  - ▶ Service is an Android Application Component that runs in the background.



- ▶ Accomplish tasks that are to be run even when none of the Activities associated with an application are visible or running.

# Application Essentials

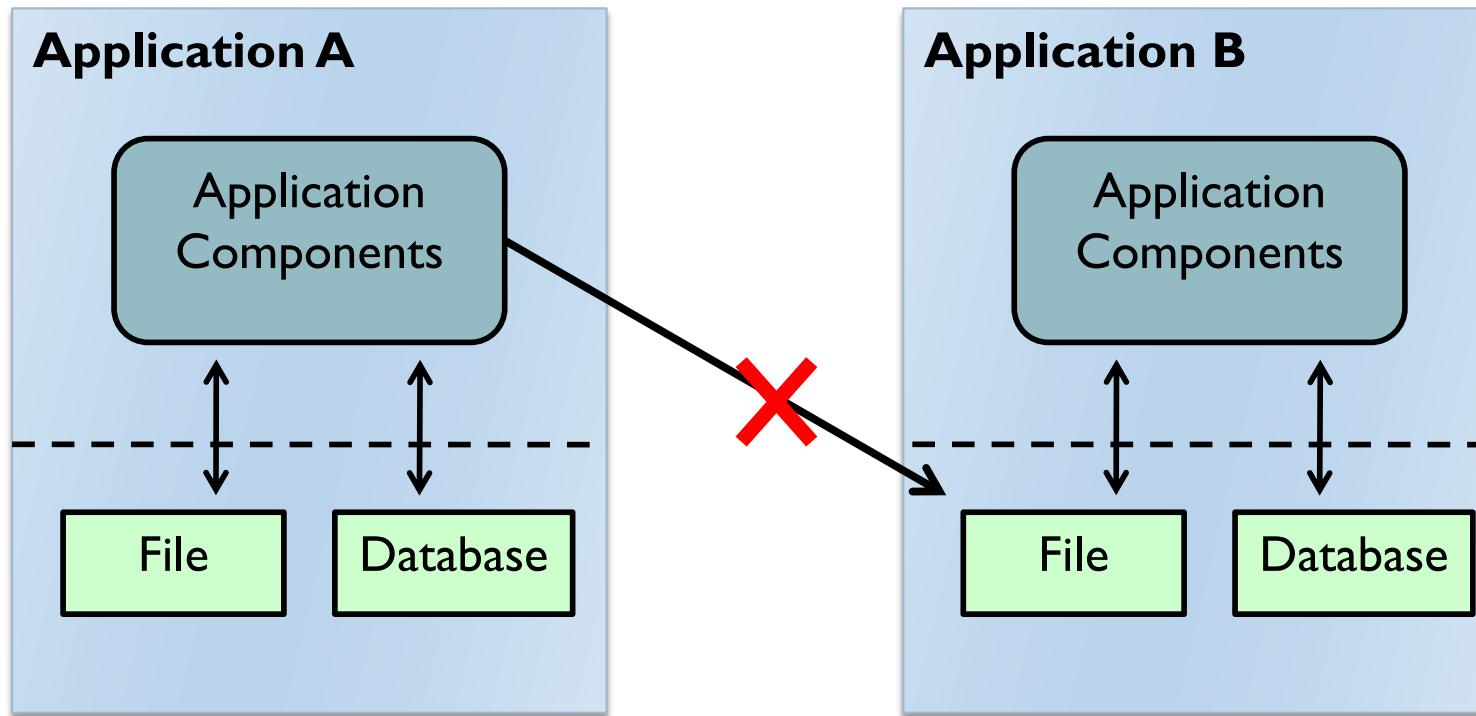
---

- ▶ **Service**
  - ▶ Service can be scheduled for execution in the background at a later point in time or periodically.
  - ▶ Common Use-cases
    - ▶ Playing music in the background
    - ▶ Syncing Application data with remote server.
    - ▶ Run scheduled tasks

# Application Essentials

## ▶ Content Provider

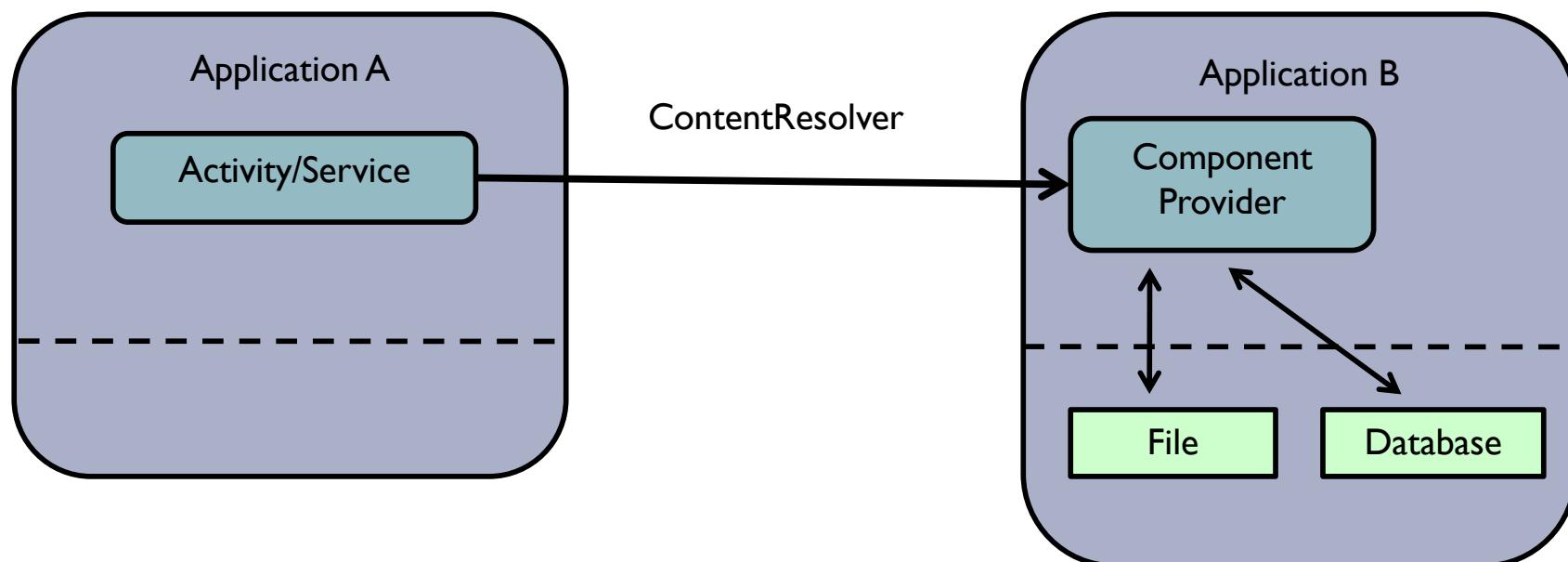
- ▶ Android Applications run in a sand-boxed environment.
  - ▶ An application does not have direct access to files and directories of other applications.



# Application Essentials

- ▶ Content Provider

- ▶ Allows sharing data between applications
  - ▶ By implementing content provider, an application can share its data with other applications



# Application Essentials

---

- ▶ **Content Provider**
  - ▶ Android ships with a number of content providers for common data
    - ▶ call log, contacts, browser information, audio and video media etc.



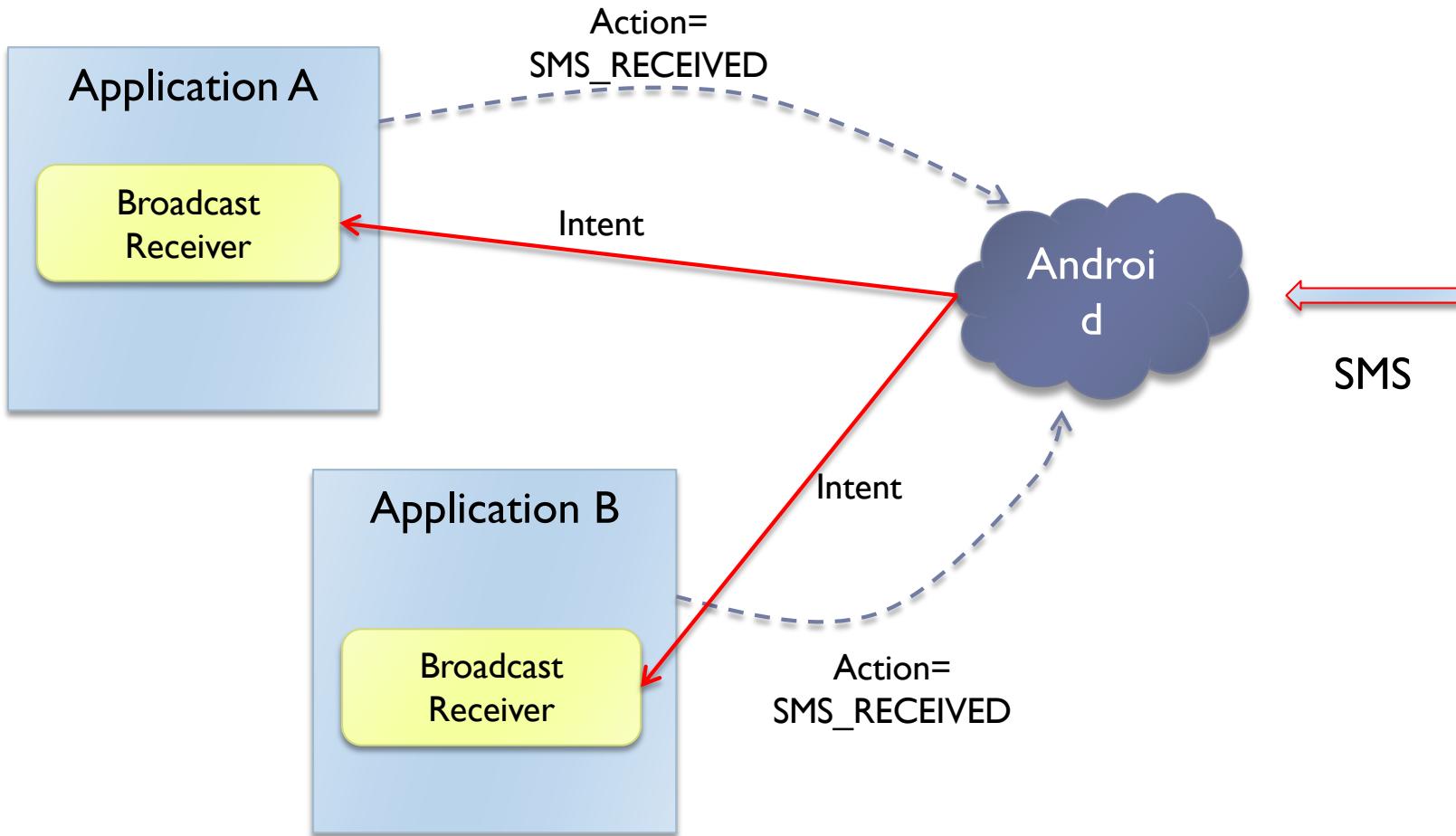
# Application Essentials

---

- ▶ **Broadcast Receiver**
  - ▶ System and Application broadcast events.
    - ▶ Android System generates broadcasts when certain state changes occur.
    - ▶ Applications can generate broadcasts.
  - ▶ Applications that want to listen for broadcasts
    - ▶ Implement Broadcast receiver component
    - ▶ Register the Broadcast Receiver for the Broadcast Action.

# Application Essentials

## ▶ Broadcast Receiver

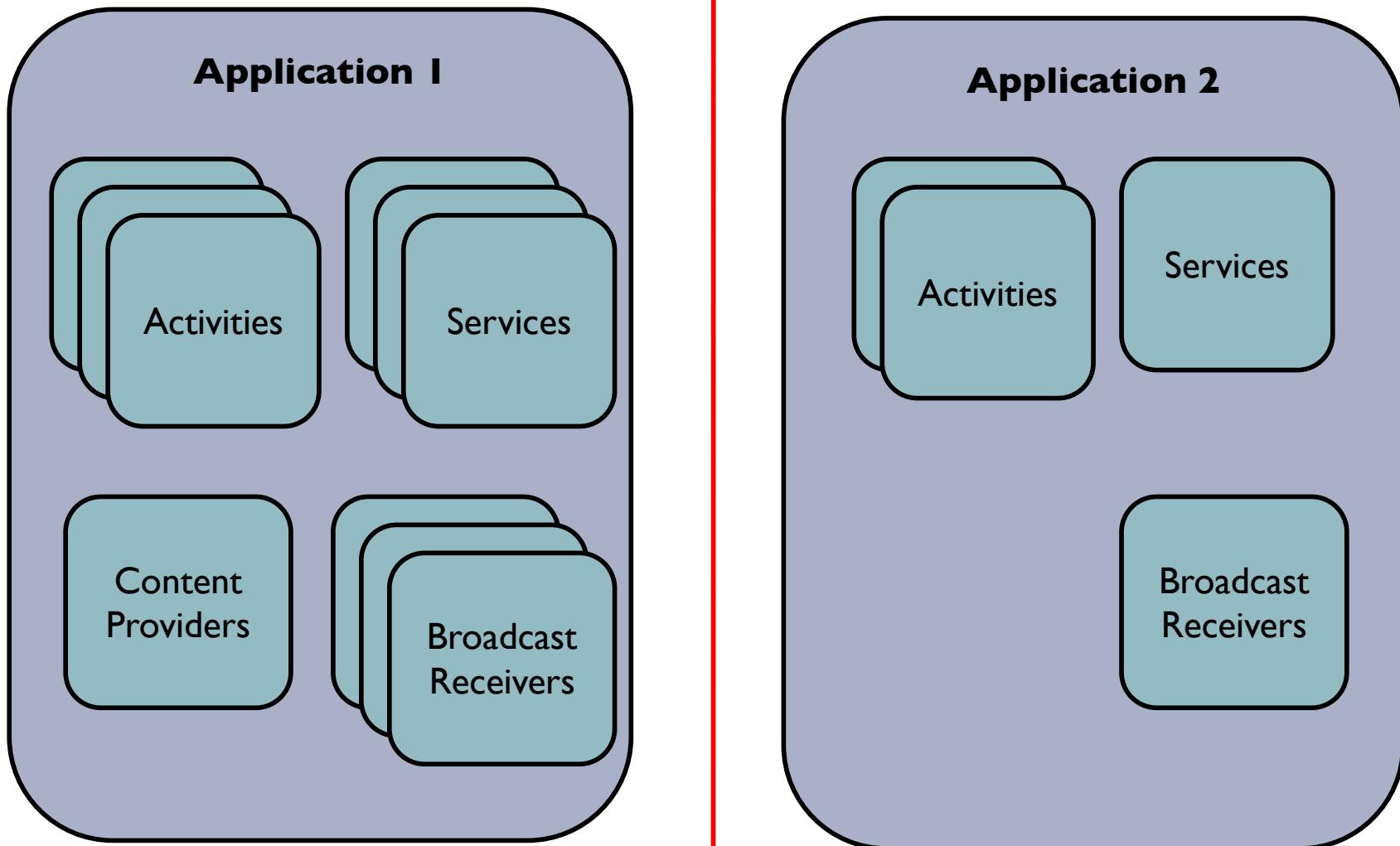


# Application Essentials

---

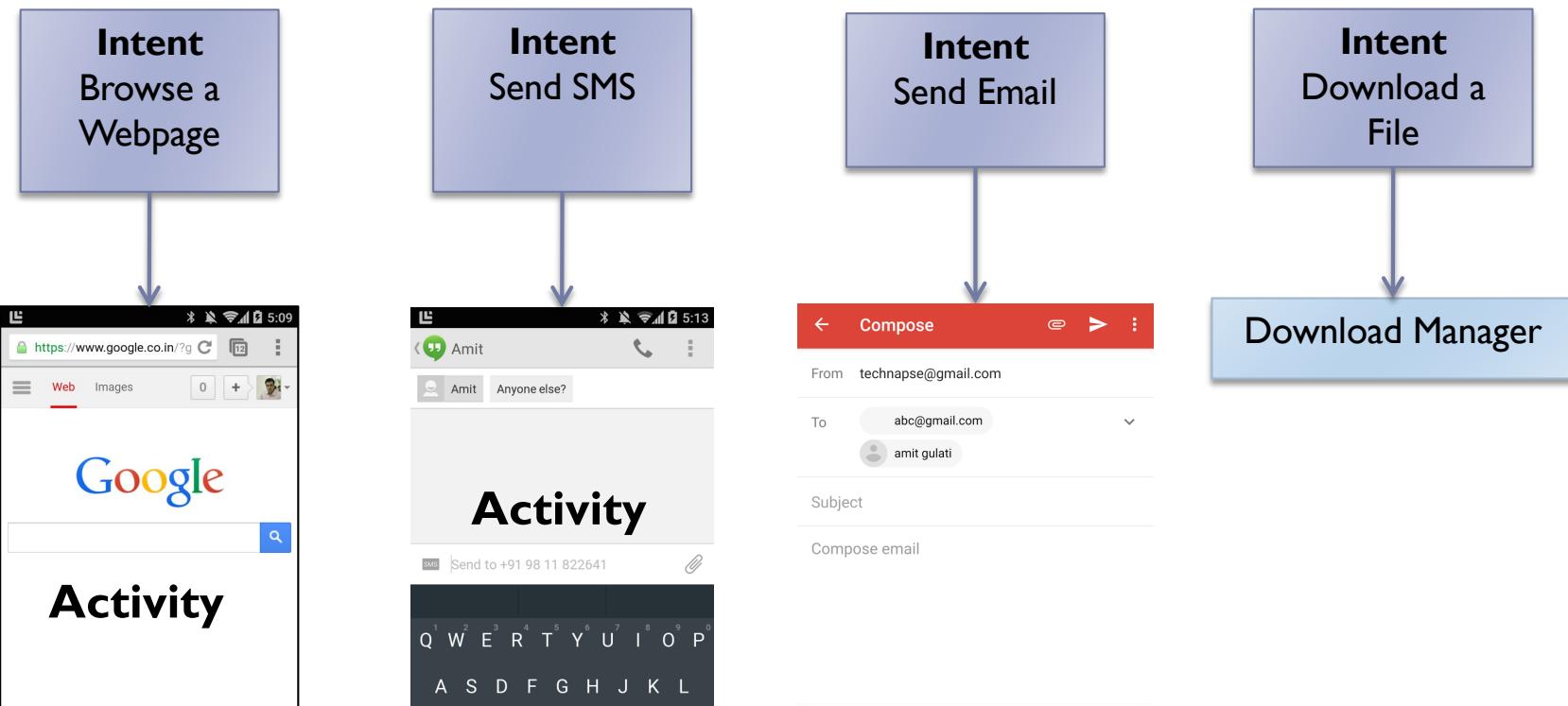
- ▶ **Broadcast Receiver**
  - ▶ Application with broadcast receiver does not have to be running or in background
    - ▶ Launched automatically in the background in order to execute Broadcast Receiver.
    - ▶ Executed as a result of a specific broadcast is raised by the System or an Application.
  - ▶ Broadcast announcements generated by system may include:
    - ▶ Airplane mode
    - ▶ Phone call received.
    - ▶ Low battery
    - ▶ USB connection
    - ▶ Etc.

# Application Essentials



# Application Essentials

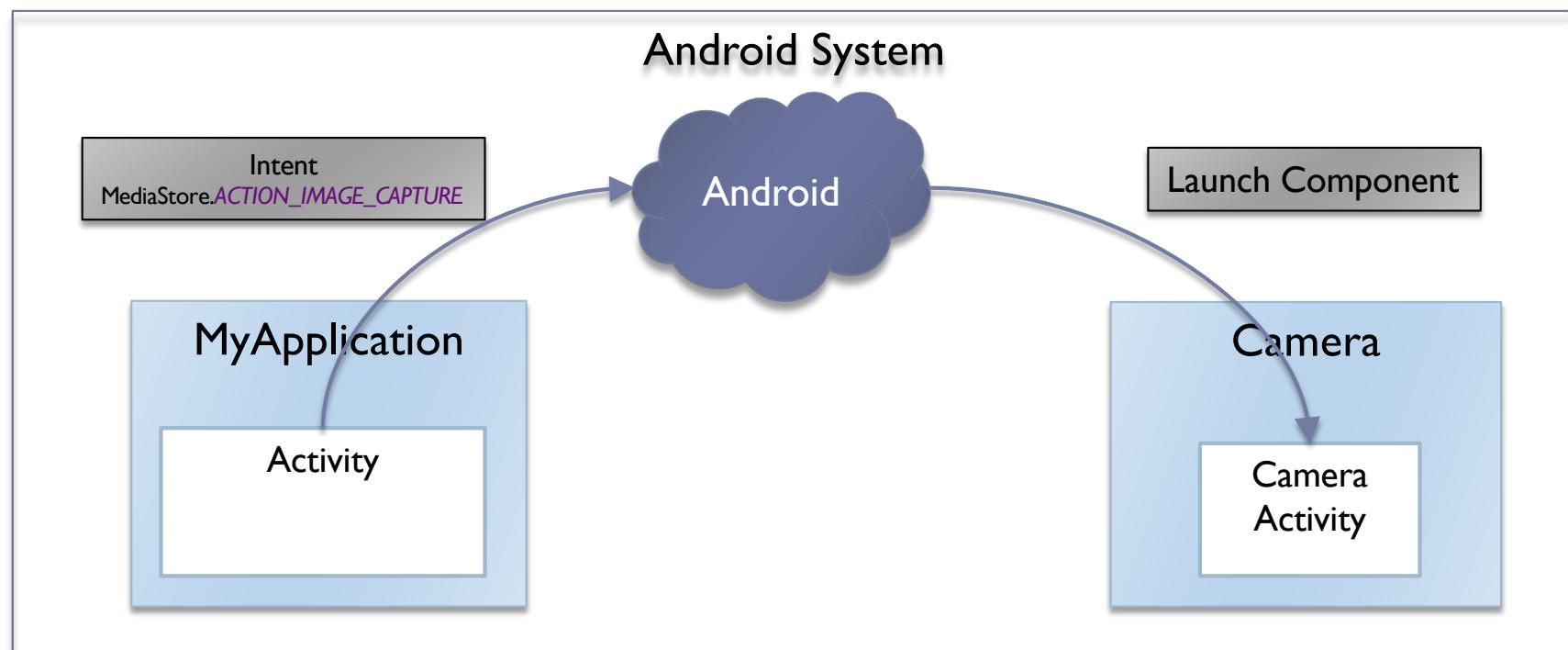
- ▶ Intent
  - ▶ Abstract concept that represent
    - ▶ Operation or action to be performed.
  - ▶ Actions are performed via Application components



# Application Essentials

## ▶ Intent

- ▶ Primary way for launching Application components.
- ▶ Components that are launched using Intent may be part of other applications



# Application Essentials

---

- ▶ Intent
  - ▶ Example

```
var intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)  
startActivity(intent)
```



# Application Essentials

---

- ▶ Intent

- ▶ Primary way for launching Application components.

- ▶ Primary way in which system / third party application send events to applications.
  - ▶ Communication between application components

# Application Essentials

---

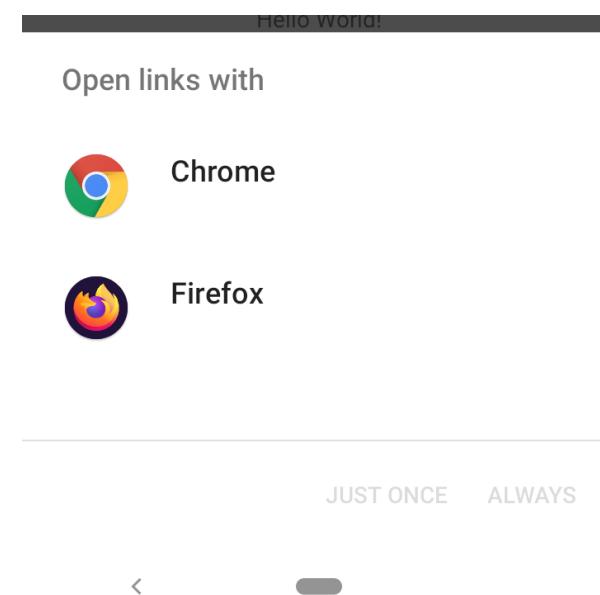
- ▶ Why Intents ?
  - ▶ Re-usability & Loose Coupling
    - ▶ Use of components written by other programmers without knowing much about them.
    - ▶ To use the Camera Activity in android

```
Intent camera =  
    new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

# Application Essentials

- ▶ Why Intents ?
  - ▶ Re-usability & Loose Coupling
    - ▶ Intents are bound to Application components at Run-time
    - ▶ In case of multiple components for an intent the user can select which one to use.

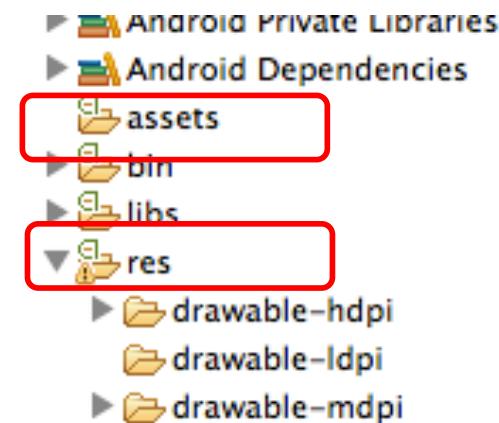
```
var intent = Intent(Intent.ACTION_VIEW)  
intent.data = Uri.parse("http://www.google.com")  
startActivity(intent)
```



# Application Essentials

---

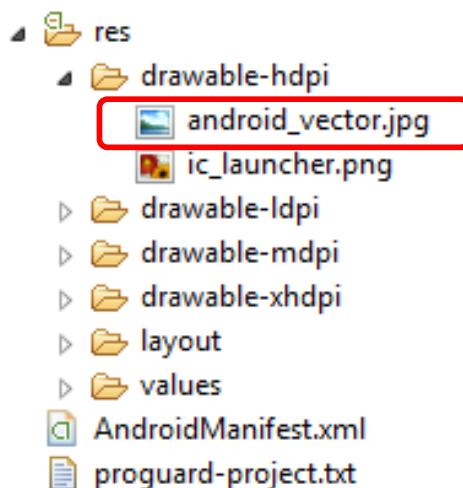
- ▶ **Android Application Resources**
  - ▶ Resources are things that are used by your application other than source code.
    - ▶ Images (Icons, Splash Screen, etc.)
    - ▶ XML files
    - ▶ Media files (Audio, Video etc.)
    - ▶ Strings, Arrays, Styles and Themes etc.
- ▶ **Two folders for adding resources**
  - ▶ **res**
  - ▶ **assets**



# Application Essentials

## ▶ Android Application Resources

- ▶ “res” folder resources
  - ▶ Android Asset Packaging Tool (AAPT) processes resources added to the “res” folder in project.
  - ▶ A unique entry (ID) for each resource is placed in the R.java file.
  - ▶ Resource can then be accessed in an Android project with the ID.

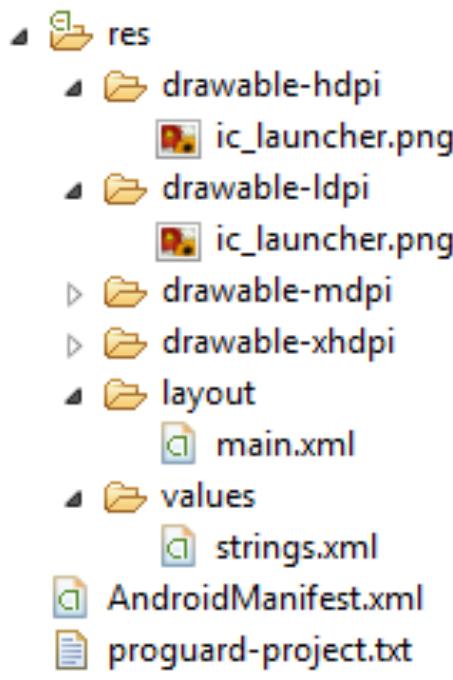


```
public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int android_vector=0x7f020000;
        public static final int ic_launcher=0x7f020001;
    }
}
```

# Application Essentials

## ▶ Android Application Resources

- ▶ Resources added to the application project are packaged together in the application bundle (.apk) that is installed on the device.



# Application Essentials

---

- ▶ **Android Manifest**
  - ▶ Every Android application package contains an Application Manifest file (`androidmanifest.xml`).
  - ▶ Contains application meta-data, that is used by Android system.
  - ▶ Application Manifest file serves the following purpose
    - ▶ Application information i.e. Name, Version, icon, etc are defined in the manifest file.
    - ▶ Used to specify hardware and software dependencies of the application.
    - ▶ Used to specify permissions required by the application.
    - ▶ Used to define the different Application components present in the application package.



# Application Essentials

## ▶ Android Manifest

- ▶ Manifest file is located at the root level in the Android project hierarchy.
- ▶ Manifest file is packaged along with executable code and resources in the .apk file.

