LA TROBE UNIVERSITY

All kinds of clever

CSE2MAD

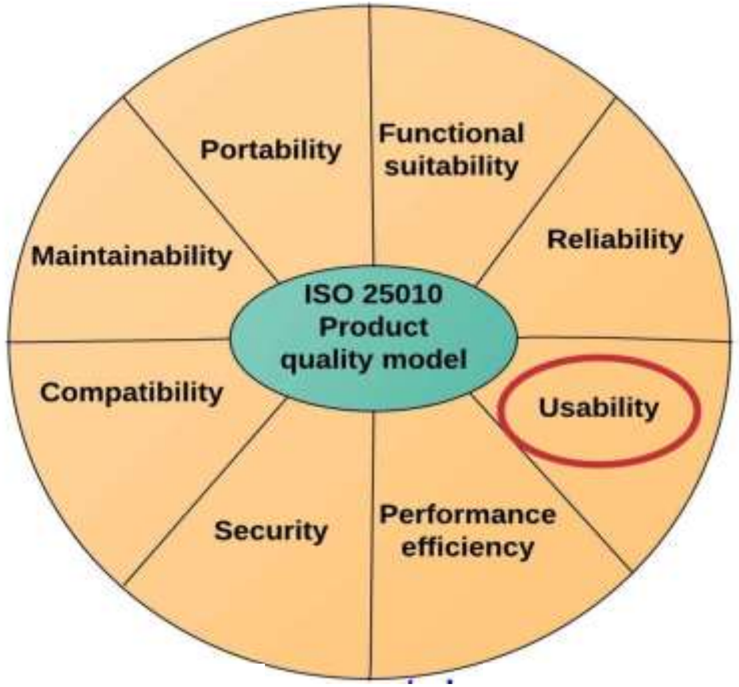
Mobile Application Development
Lecture 2

latrobe.edu.au   CRICOS Provider 00115M   Updated 2020 by Marita Fitzgerald

User-Centered Design Concepts for Mobile Applications

What is Usability?

https://youtu.be/dln9xDsmCoY



https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en

Usability is one of the eight ISO software quality attributes

Usability is …

The degree to which a product or system can be used by **specified users** to achieve **specified goals** with effectiveness, efficiency and satisfaction in a **specified context of us**e.

- Appropriateness recognizability
- Learnability
- Operability
- User error protection
- User interface aesthetics
- Accessibility

https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=3&start=3

2.1.1 Mobile App development vs Desktop app development

Is a typical PC application different than a typical mobile application?

Modes of interaction

Power

Mobility

CPU, Memory, Storage

CD-ROM/DVDs & portable external storage

Sensors

Connectivity

Interaction: Desktop

- Typical monitor – at least 17 inches, 1,280 x 960 pixels resolution
- Full keyboard - provides keys to all Characters, distance between keys on about 19 mm

WIMP: windows, icons, menus, and pointers
Plus other I/O h/w such as joysticks, cameras, microphones etc.

Interaction: Mobile device

- Screens of various sizes, densities, and specifications
- Screen size is smaller. Eg: Galaxy s7 display is 5.1”
- Different aspect ratios (the ratio of the width to the height)
- Keypad instead of the full keyboard/ Onscreen keyboard
- Touch screens and styluses, instead of the mouse

UI design considerations for mobile devices

Consider the following:

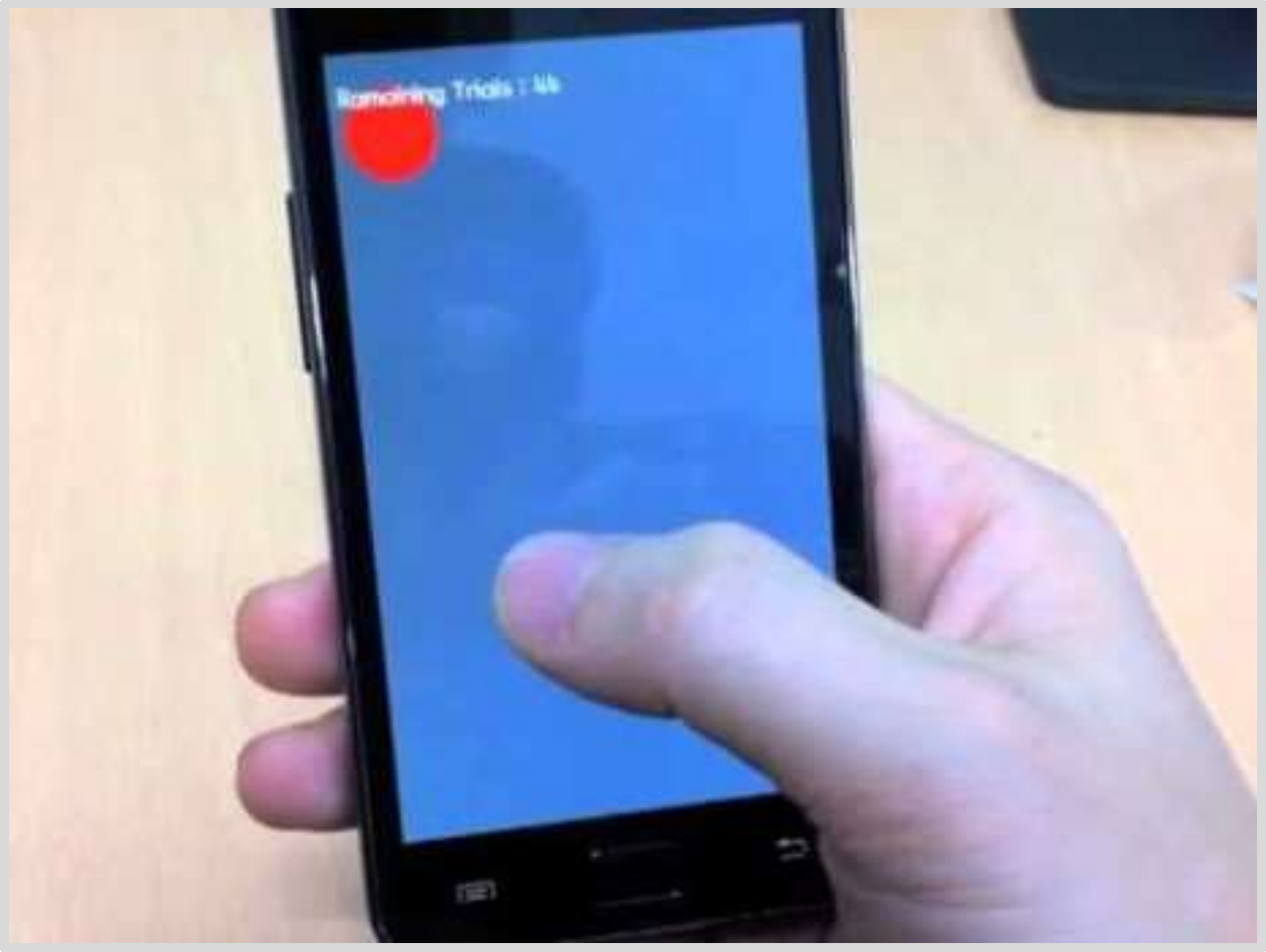
- Screen size of the mobile device/s & Size of application windows
- Touch screens and styluses
- Keyboard input problems
- S/W distribution & copyright

UI design considerations for mobile devices (cont..)

1. Screen size of the mobile device/s & Size of application windows
   - Size of text and Icons
   - Clickability of Buttons and other Graphical Elements

2. Touch screens and styluses cannot provide all mouse functions.
   - No right button
   - Current finger/stylus location cannot be captured
   - when the screen is not touched.

Unreachable objects example



https://youtu.be/eBXzwAD06fs

UI design considerations for mobile devices (cont..)

3. Keyboard input problems

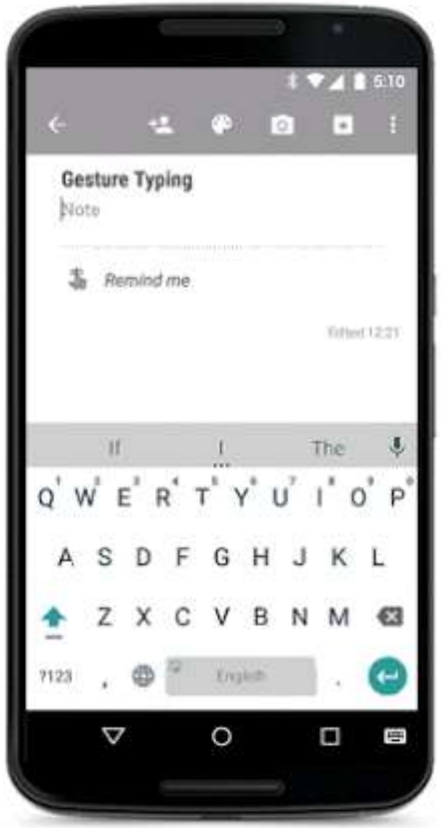
- PCs have full keyboards, whereas mobile devices usually have simpler keypads
- “Fat Finger” problem for touchscreens

Approaches:

Restrict character input

Context-sensitive buttons (buttons that appear only when

Gesture Typing eg: Swype, Google keyboard 3eMs



UI design considerations for mobile devices (cont..)

4. S/W distribution & copyright:

Unlike PCs mobile devices such as smartphones and tablets do not have CD-ROM/DVD drives.

Solution is to have ‘online stores’ such as the Apple Store or Google Play.

View the article below for some expended Do’s and Don’ts of mobile UI Design

https://xd.adobe.com/ideas/principles/app-design/10-dos-donts-mobile-app-design/

2.1.3 Fitts' law

- Named after Paul Fitts, for his 1954 study of pointing.
- Predicts the time a human needs to point at a target of given size in a given distance.
- Applies to mouse/pointing movements.

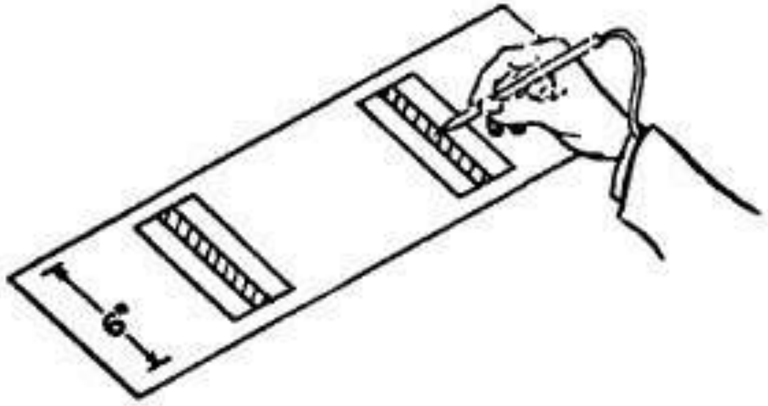
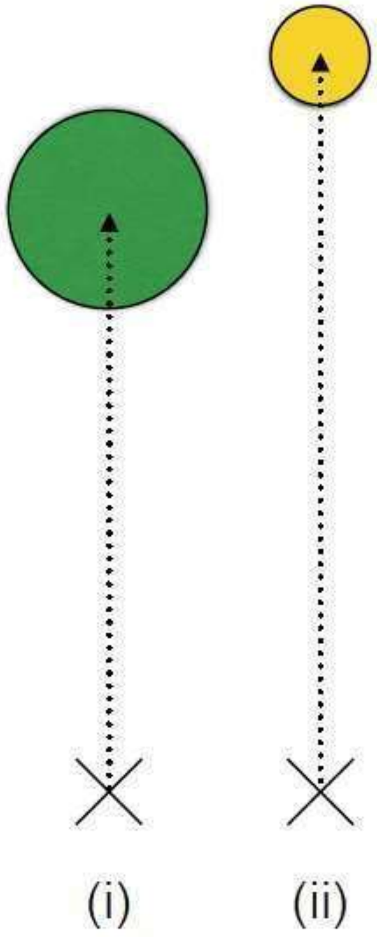


Figure 1. Reciprocal tapping apparatus. The task was to hit the center plate in each group alternately without touching either side (error) plate.

Fitts, Paul M. "The information capacity of thehuman motor system in controlling the amplitude of movement." *Journal of experimental psychology*47.6 (1954): 381.

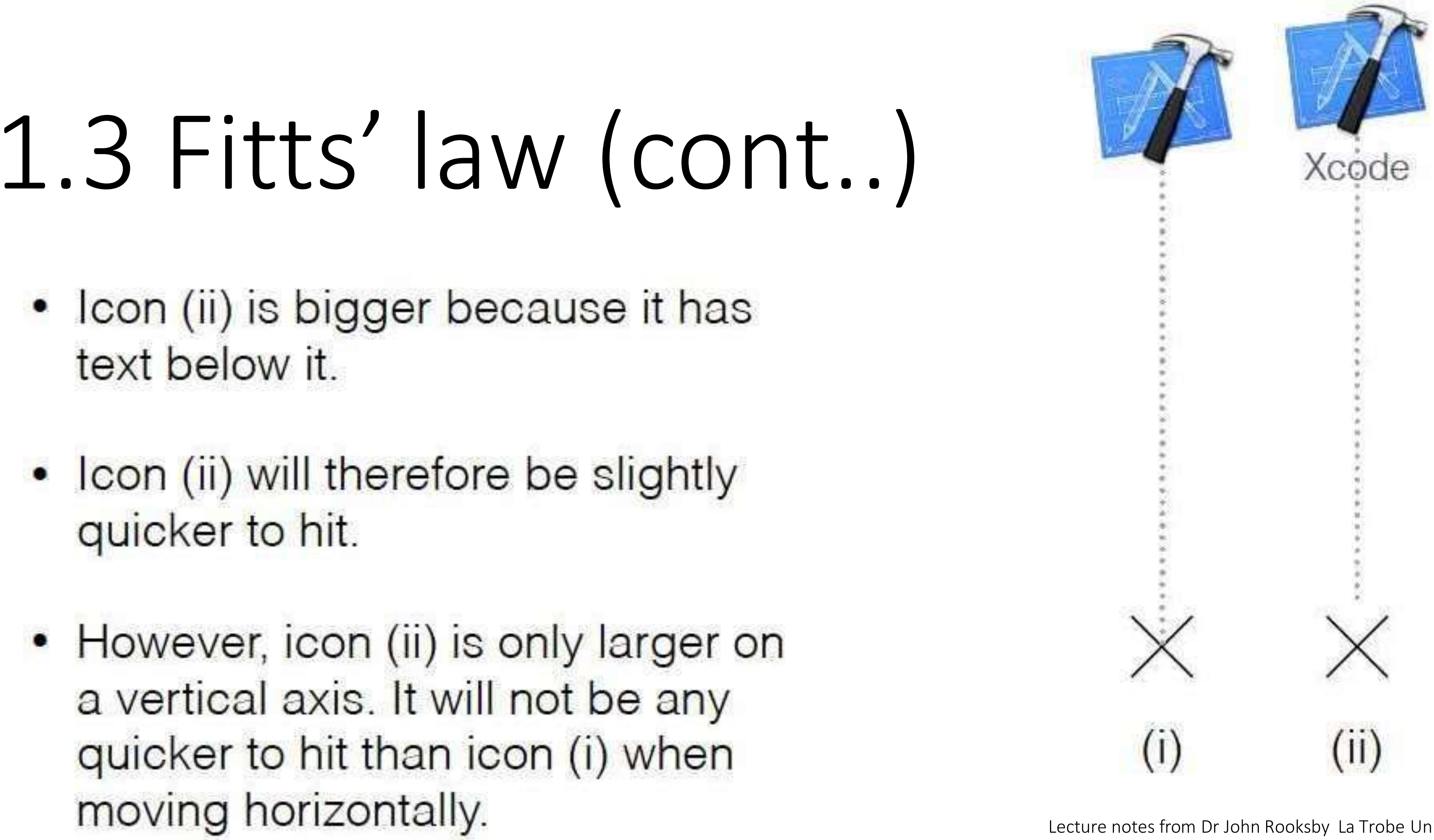
2.1.3 Fitts' law (cont..)

- Fitts' law tells us how long it will take to move a pointer from a specific position to hit different targets.
- Targets that are larger and closer are easier to hit than ones that are smaller and further away.



Lecture notes from Dr John Rooksby La Trobe University

2.1.3 Fitts’ law (cont..)



- Icon (ii) is bigger because it has text below it.
- Icon (ii) will therefore be slightly quicker to hit.
- However, icon (ii) is only larger on a vertical axis. It will not be any quicker to hit than icon (i) when moving horizontally.

Lecture notes from Dr John Rooksby La Trobe University

Fitts’ law for mobile devices?

How large should the button be?

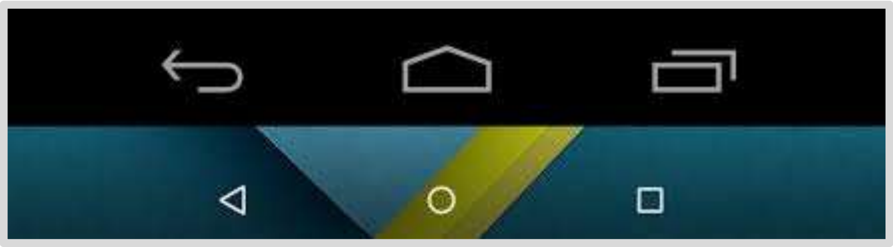
Where should you put the icons?

How easy is it to click on the menu? etc

Eg: Place high risk targets (close, delete, etc.) in relatively hard to use areas of the screen.
Place low risk highly used targets in easy to use areas. Place the high-risk targets away from the low-risk targets in order to help prevent user error.

Use it as a guideline when designing your interfaces, not as a law.

A Typical Android Device UI



◁

The BACK button takes you back to the previous screen. If the App has one screen, this will quit the app.

○

The HOME button takes you to the Home screen of your phone.

□

The OVERVIEW button will show you a list of recent apps.

https://support.google.com/android/answer/9079644

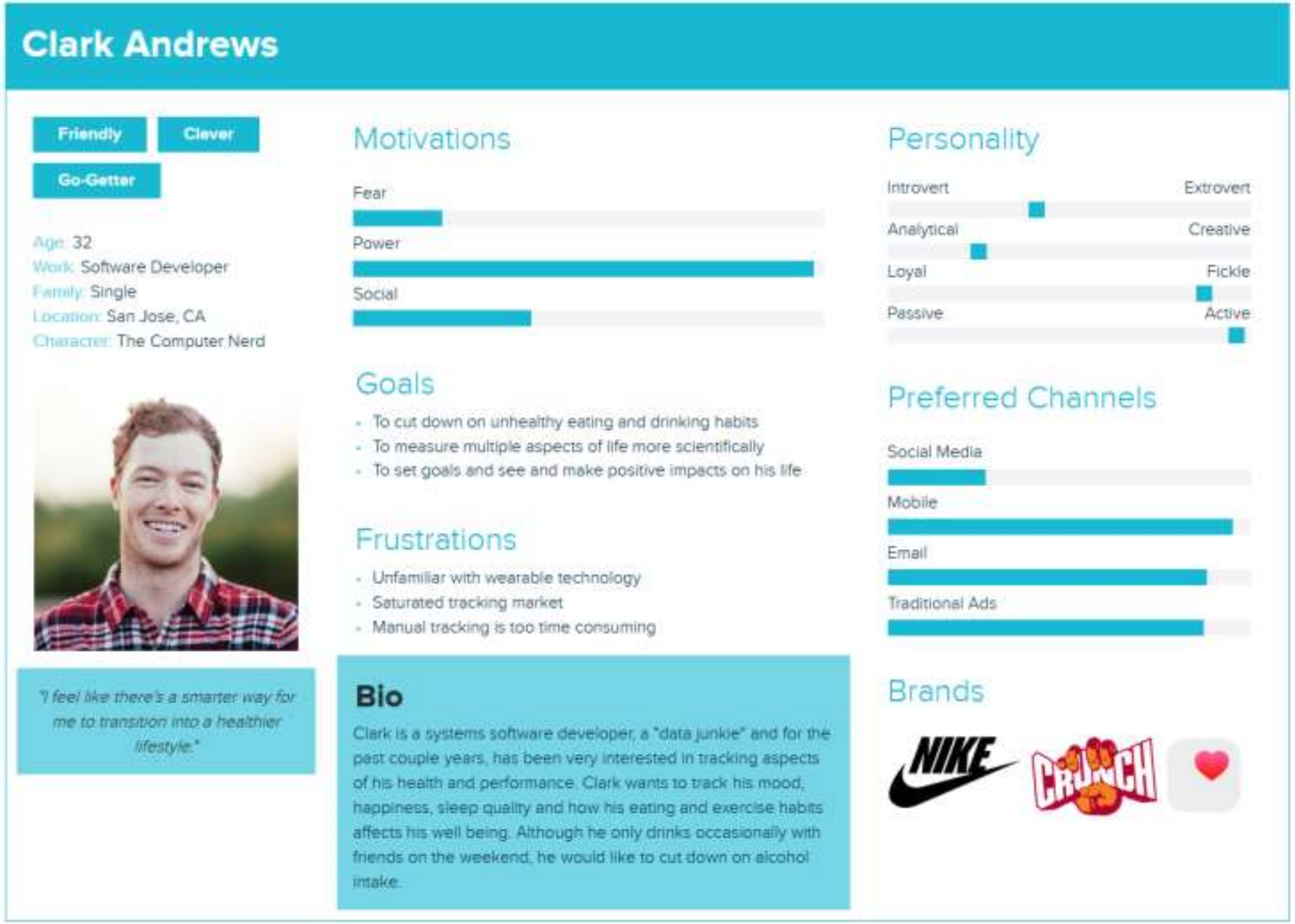
2.1.4 Interaction Design

- Interaction design is not Interface design.
- Interaction design has more to do with the design concepts and interface tools used to present information to a user.
- understand and influence users’ behaviors
- focused on the user’s reaction or on habits that develop in response to GUI elements
- Interaction design isn’t limited to mobile applications, everything we use involves interaction design

Who are you designing for?

- Who will use your app?
  - May be people like yourself – but not entirely
  - Most apps won’t be marketable or functional for every single mobile device owner.

Who are you designing for? ctd…



Clark Andrews

Friendly Clever Go-Getter

Age: 32
Work: Software Developer
Family: Single
Location: San Jose, CA
Character: The Computer Nerd

"I feel like there's a smarter way for me to transition into a healthier lifestyle."

Motivations

Fear

Power

Social

Goals

- To cut down on unhealthy eating and drinking habits
- To measure multiple aspects of life more scientifically
- To set goals and see and make positive impacts on his life

Frustrations

- Unfamiliar with wearable technology
- Saturated tracking market
- Manual tracking is too time consuming

Bio

Clark is a systems software developer, a "data junkie" and for the past couple years, has been very interested in tracking aspects of his health and performance. Clark wants to track his mood, happiness, sleep quality and how his eating and exercise habits affects his well being. Although he only drinks occasionally with friends on the weekend, he would like to cut down on alcohol intake.

Personality

Introvert Extrovert

Analytical Creative

Loyal Fickle

Passive Active

Preferred Channels

Social Media

Mobile

Email

Traditional Ads

Brands

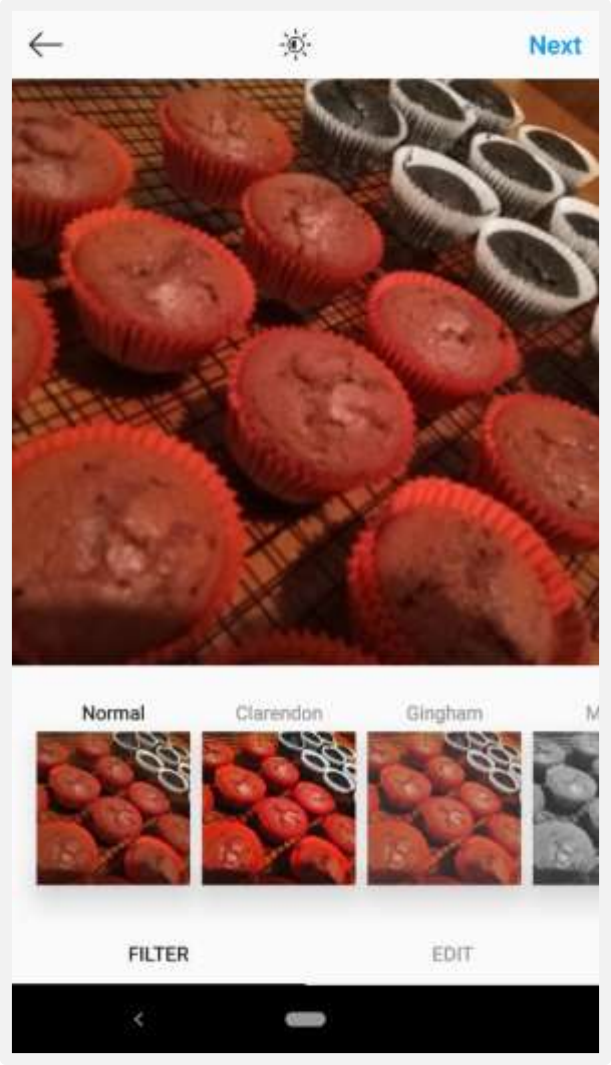
https://xtensio.com/how-to-create-a-persona/

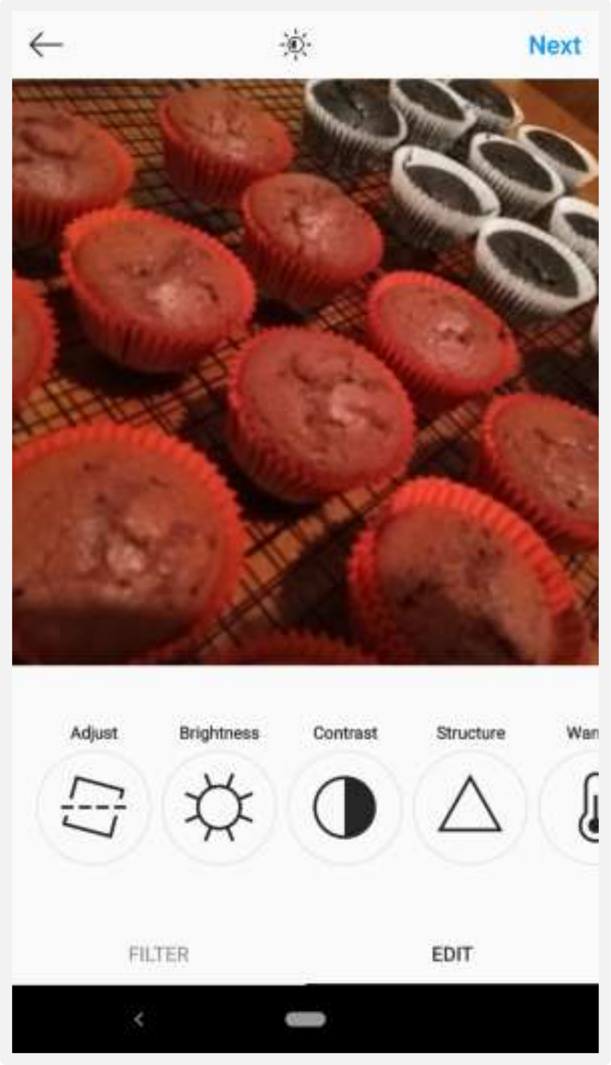
The purpose of personas is to create reliable and realistic representations of your key audience segments for reference.

These representations should be based on qualitative and some quantitative user research and web analytics.

https://www.usability.gov/how-to-and-tools/methods/personas.html

Who are you designing for? ctd…





When building apps, create experiences that give the user a sense of security to explore your work. Eg Instagram

Mobile user work-flow

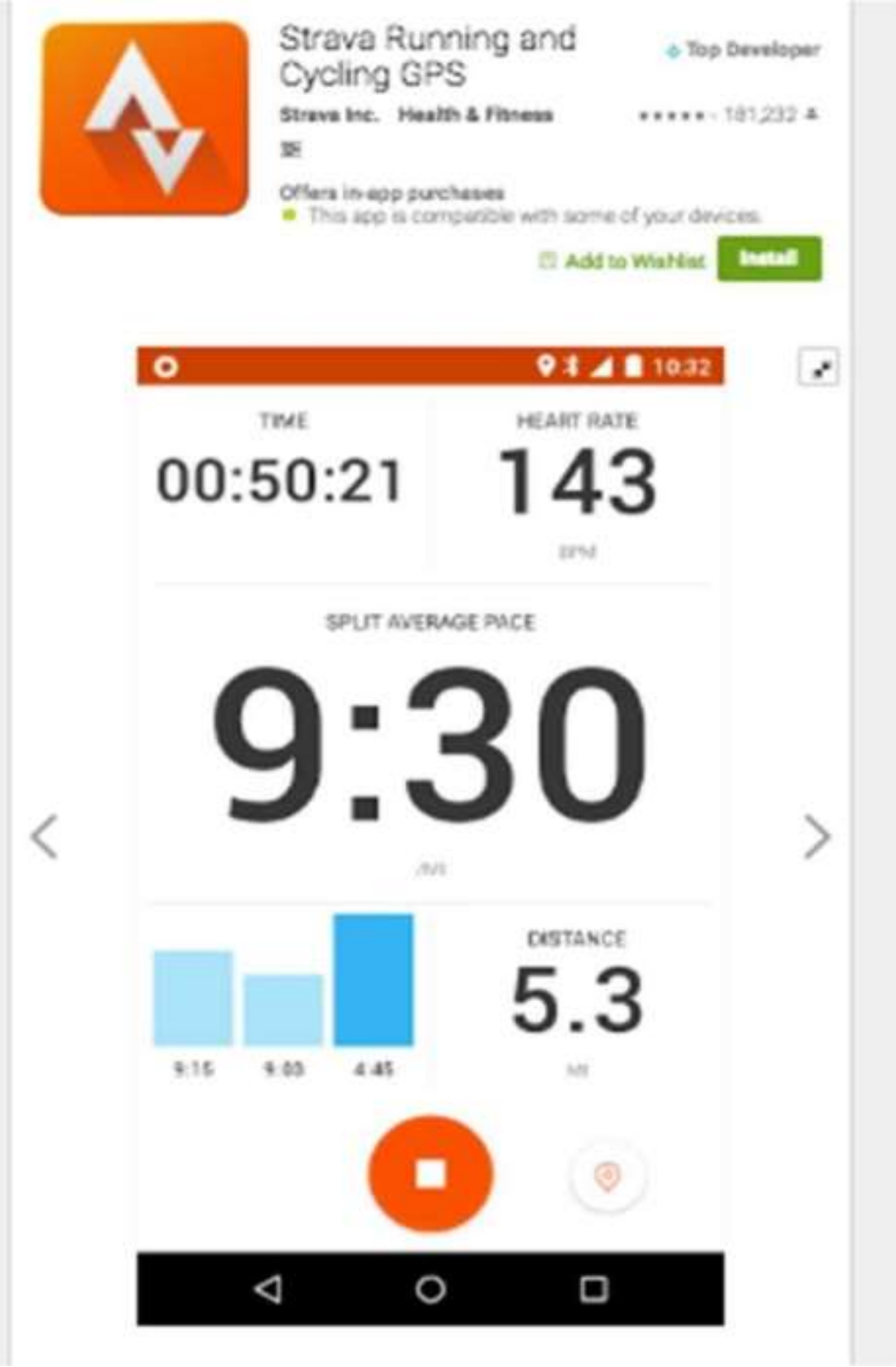
- Mobile apps typically cost very little (~<$5), so users will not be prepared to invest a lot of time learning how to use them. Also, mobile users have thousands of apps to choose from.
- -> easy-to-understand interfaces and user interactions that are readily apparent and require little thought.

- Consider the many situations your app will be used in. Mobile apps are typically used in different contexts than desktop applications.
- Eg: Navigators, Shopping lists, Exercise

Mobile app context of use: A cycling app

Who are the potential users & how would you make your app usable?

Mobile app context of use example: Strava



Strava Running and Cycling GPS

Strava Inc. Health & Fitness

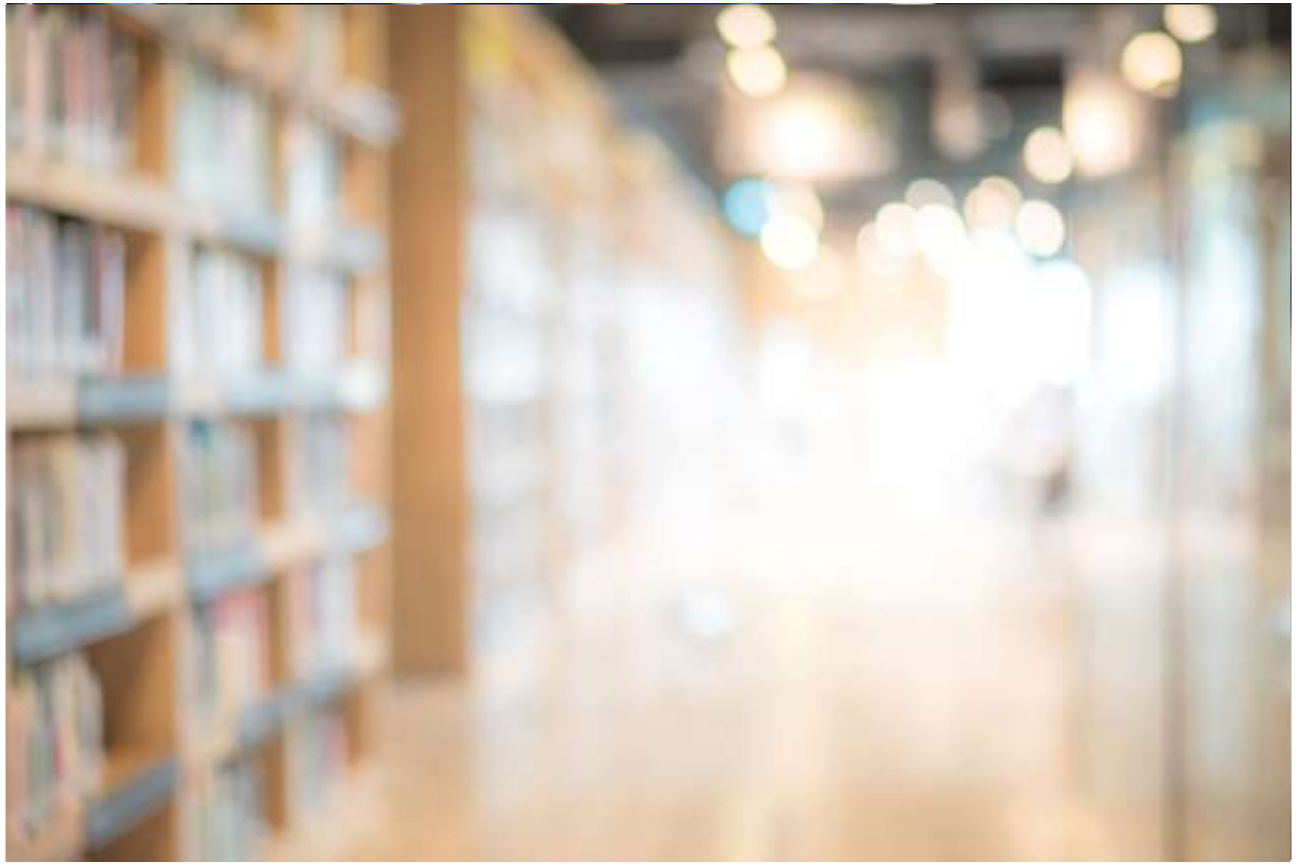
Offers in-app purchases

This app is compatible with some of your devices.

https://apps.apple.com/us/app/strava-run-ride-swim/id426826309

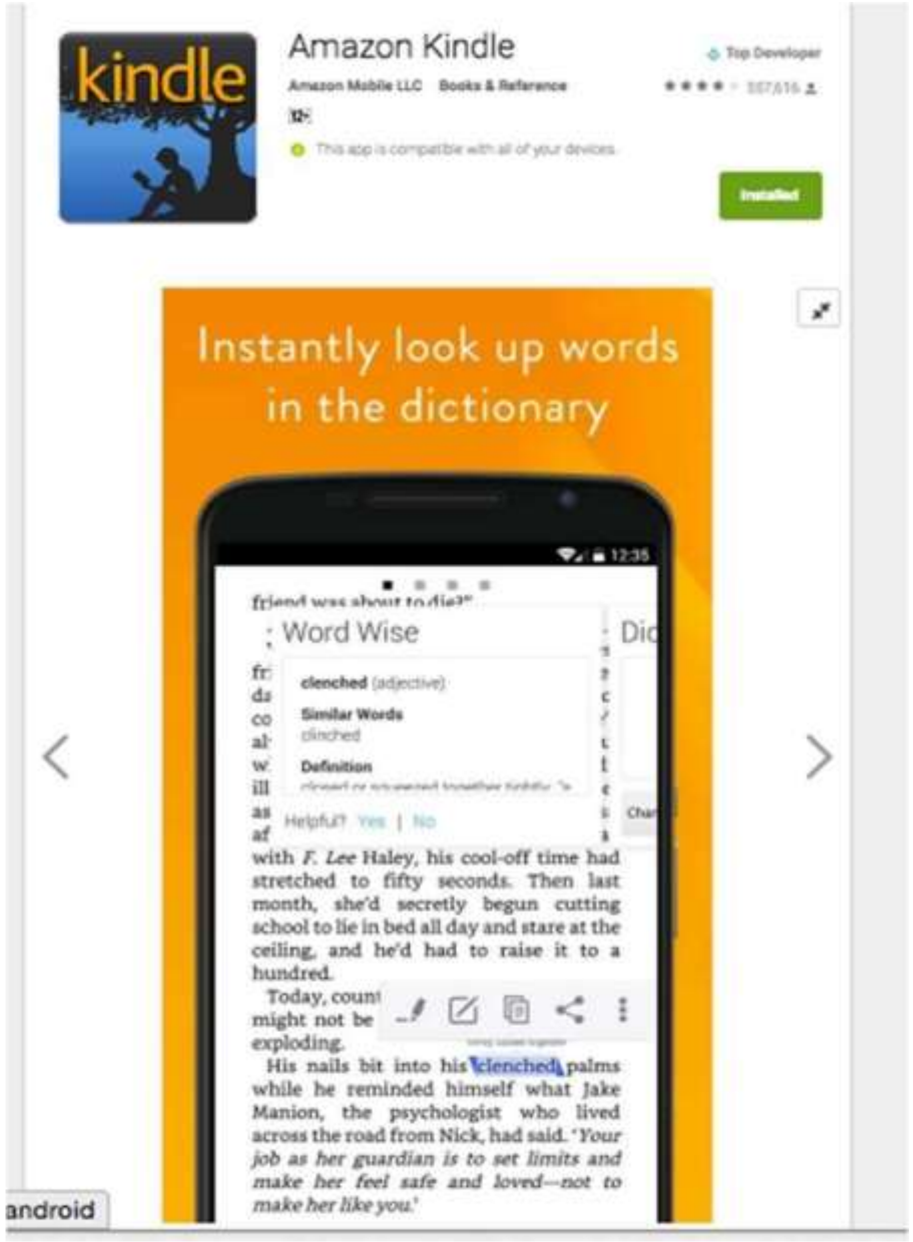
What has the designer done to ensure that this app will provide the best experience in its context of use?

Mobile app context of use: A reading app



Who are the potential users & how would you make your app usable?

Mobile app context of use example: Kindle



friend was about to die?"

with *F. Lee* Haley, his cool-off time had stretched to fifty seconds. Then last month, she'd secretly begun cutting school to lie in bed all day and stare at the ceiling, and he'd had to raise it to a hundred.

Today, count might not be exploding.

His nails bit into his clenched palms while he reminded himself what Jake Manion, the psychologist who lived across the road from Nick, had said. '*Your job as her guardian is to set limits and make her feel safe and loved—not to make her like you.*'

What has the designer done to ensure that this app will provide the best experience in its context of use?

https://play.google.com/store/apps/details?id=com.amazon.kindle&hl=en_AU

Reading

- Kangas, E., and T. Kinnunen. ”Applying User-Centered Design to Mobile Application Development." *Commun ACM* 48, no. 7(2005): 55-59.

- GUI design for android apps *by* Ryan Cohen 2014: Chapter 1

- Essential Mobile Interaction Design: Perfecting Interface Design in Mobile Apps *by* Cameron Banga & Josh Weinhold: Chapters 2 & 3

- https://developer.android.com/design/index.html

- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. Journal of experimental psychology.

Introduction to Android

Android History

Android Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White. Rubin described the Android project as "tremendous potential in developing smarter mobile devices that are more aware of its owner's location and preferences".

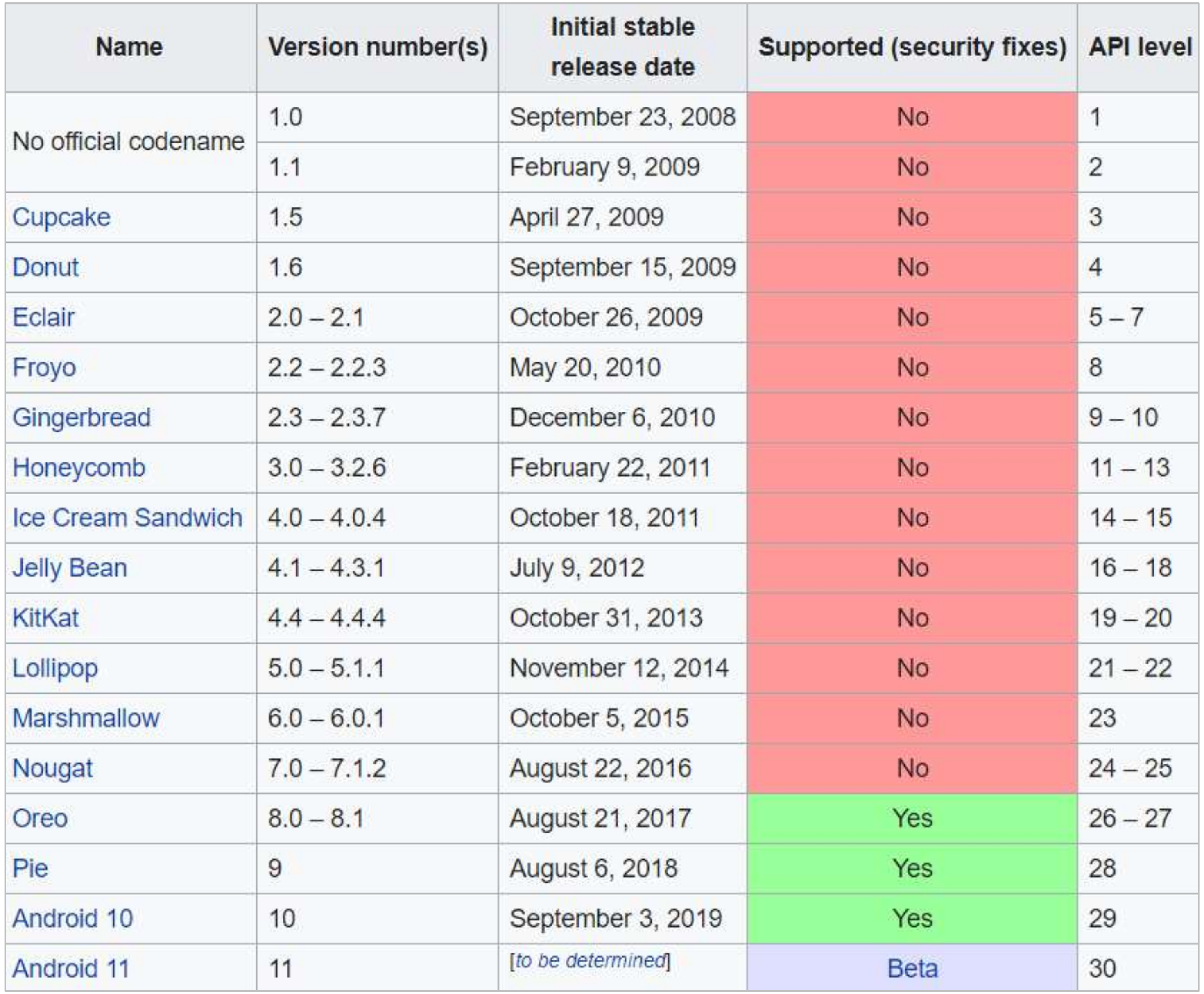
The early intentions of the company were to develop an advanced operating system for digital cameras, and this was the basis of its pitch to investors in April 2004.

The company then decided that the market for cameras was not large enough for its goals, and by five months later it had diverted its efforts and was pitching Android as a handset operating system that would rival Symbian and Microsoft Windows Mobile.

Luis Alberto Arjona Chin / CC BY (https://creativecommons.org/licenses/by/2.0)

The HTC Dream or T-Mobile G1, the first commercially released device running Android (2008)

Android version history



Name	Version number(s)	Initial stable release date	Supported (security fixes)	API level
No official codename	1.0	September 23, 2008	No	1
	1.1	February 9, 2009	No	2
Cupcake	1.5	April 27, 2009	No	3
Donut	1.6	September 15, 2009	No	4
Eclair	2.0 – 2.1	October 26, 2009	No	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	No	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	No	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	No	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	No	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	No	16 – 18
KitKat	4.4 – 4.4.4	October 31, 2013	No	19 – 20
Lollipop	5.0 – 5.1.1	November 12, 2014	No	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	No	23
Nougat	7.0 – 7.1.2	August 22, 2016	No	24 – 25
Oreo	8.0 – 8.1	August 21, 2017	Yes	26 – 27
Pie	9	August 6, 2018	Yes	28
Android 10	10	September 3, 2019	Yes	29
Android 11	11	[*to be determined*]	Beta	30

https://en.wikipedia.org/wiki/Android_version_history

Android is Java…
Dalvik virtual machine < 4.4 KitKat

- Dalvik is a virtual machine (VM) designed and written by Dan Bornstein at Google.
- Your code gets compiled into machine independent instructions called bytecodes, which are then executed by the Dalvik VM (Just-In-Time) the mobile device.
- Although the bytecode formats are a little different, Dalvik is essentially a Java virtual machine optimized for low memory requirements. It allows multiple VM instances to run at once and takes advantage of the underlying operating system (Linux) for security and process isolation.

[SRC: "Hello, Android: Introducing Google's Mobile Development Platform" (2nd edition), Ed Burnette]

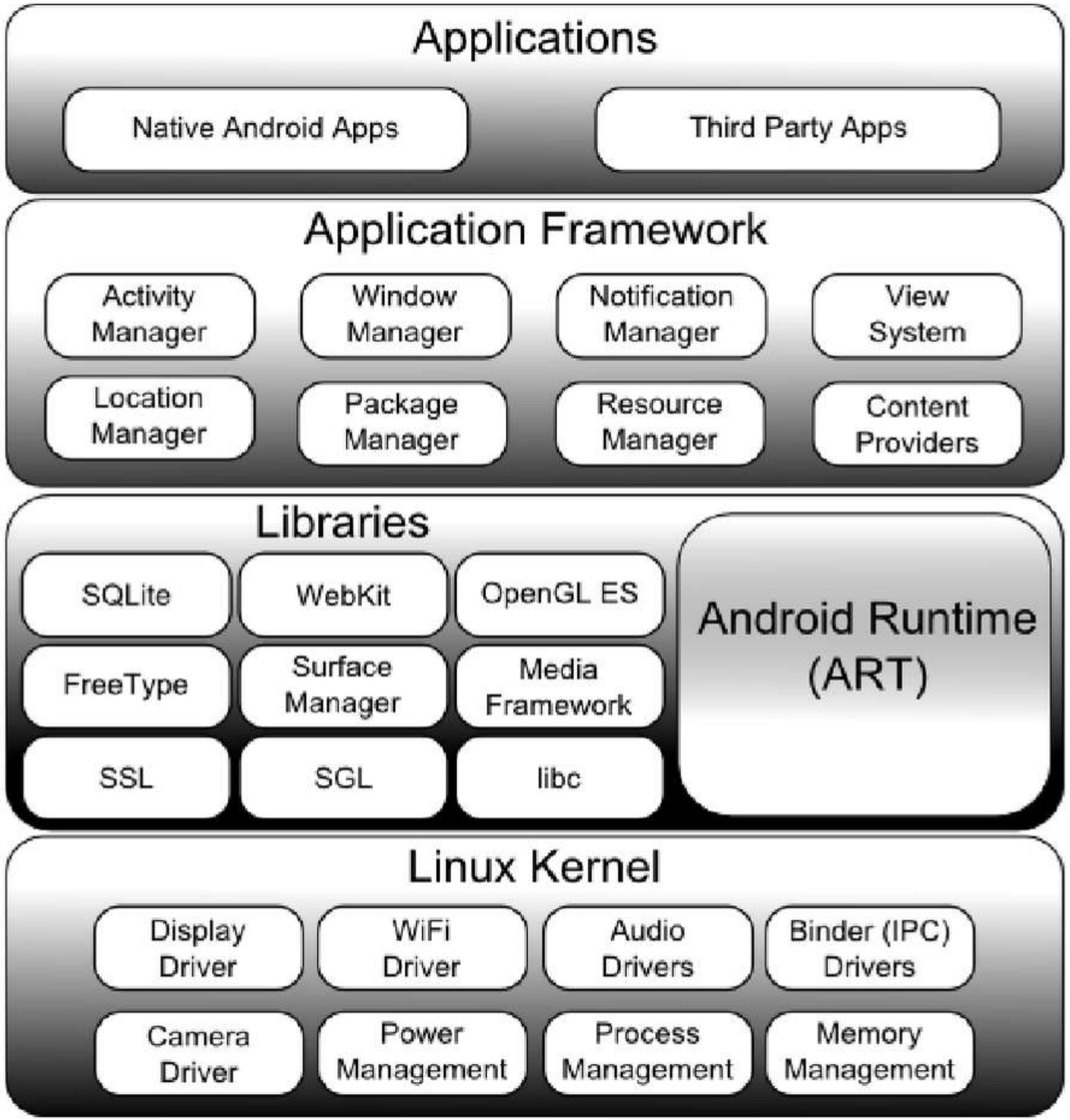
ART (the newer runtime from 5.0)

- Ahead-of-time (AOT) and just-in-time (JIT) compilation
- Optimized garbage collection (GC)
- Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields

ART Basics

https://youtu.be/USgXkI-NRPo?t=16

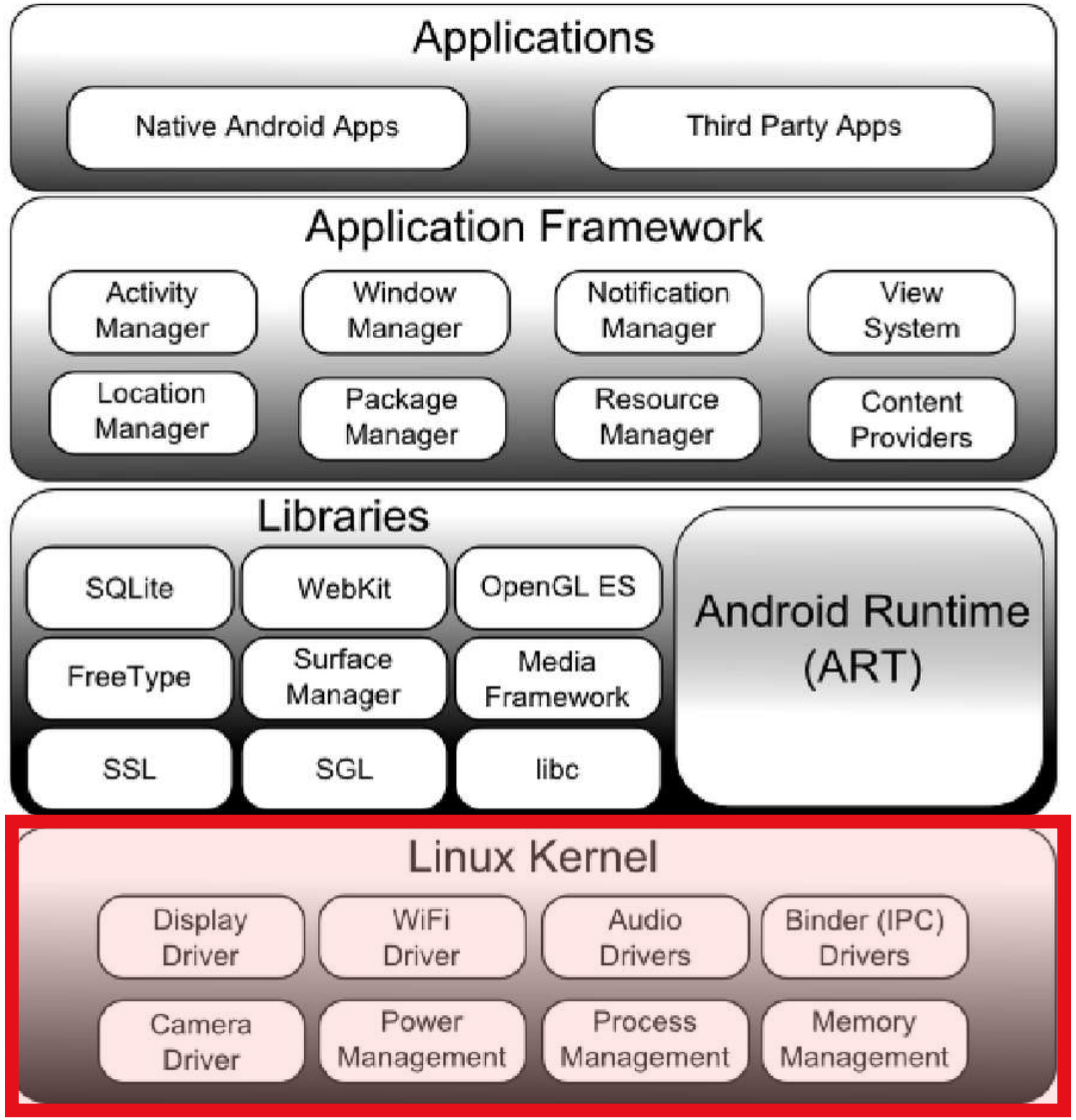
Android's layered architecture



Smyth, Neil. *Android Studio 4. 0 Development Essentials - Java Edition : Developing Android Apps Using Android Studio 4. 0, Java and Android Jetpack*

Linux Kernal

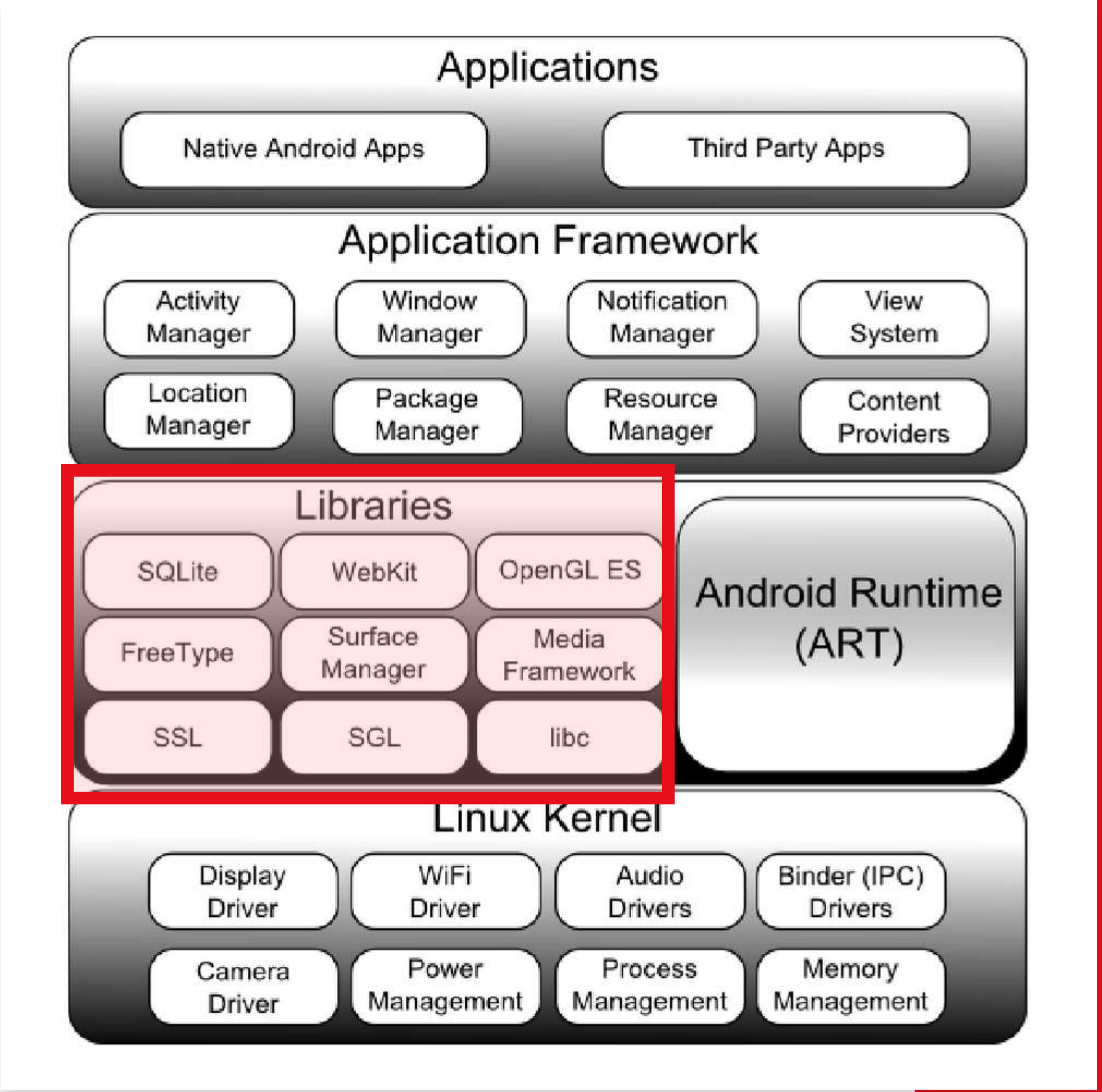
The kernel provides preemptive multitasking, low-level core system services such as memory, process and power management in addition to providing a network stack and device drivers for hardware such as the device display, Wi-Fi and audio.



Smyth, Neil. *Android Studio 4. 0 Development Essentials - Java Edition : Developing Android Apps Using Android Studio 4. 0, Java and Android Jetpack*

Libraries

These are a set of Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.

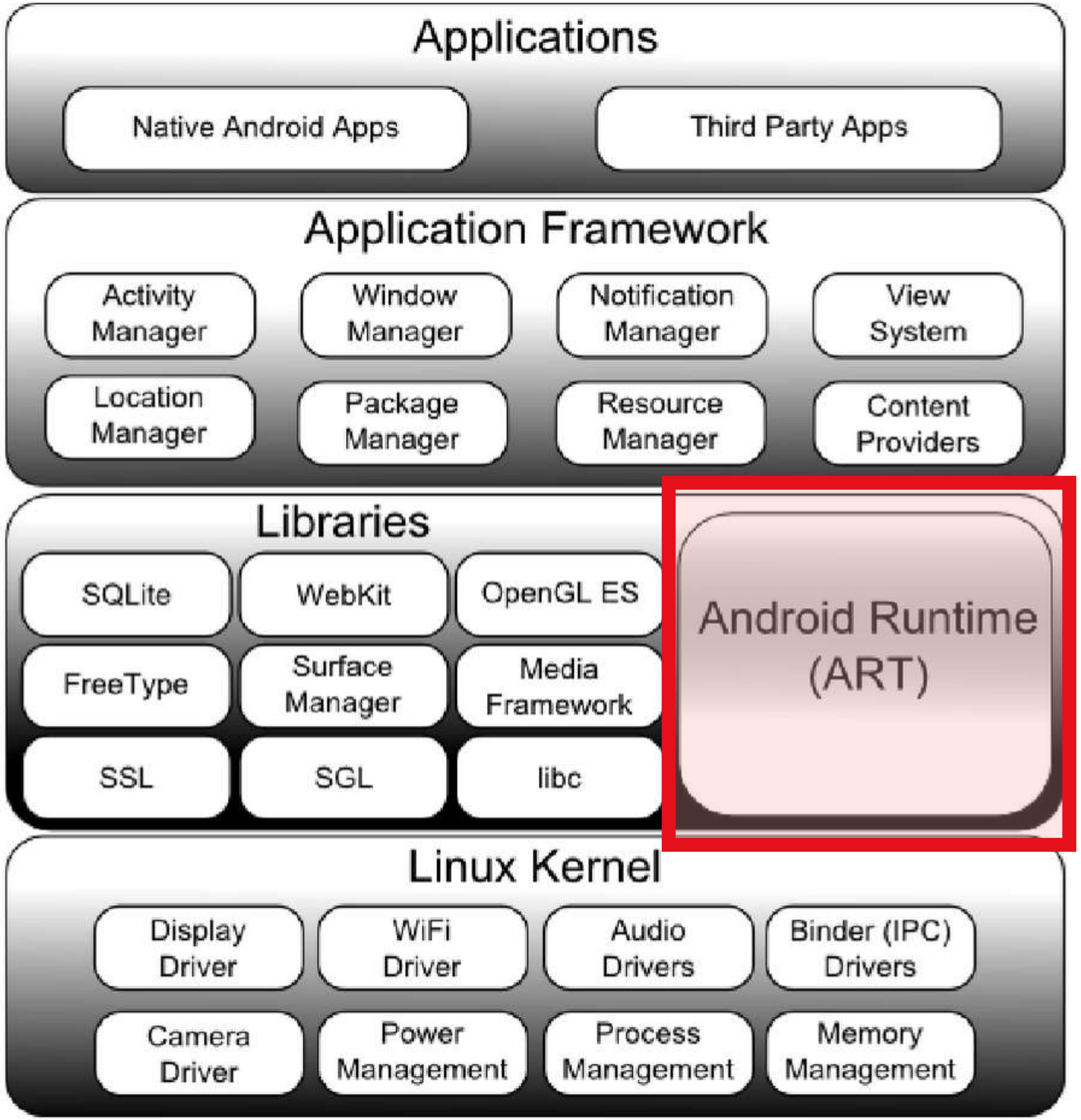


Smyth, Neil. *Android Studio 4. 0 Development Essentials - Java Edition : Developing Android Apps Using Android Studio 4. 0, Java and Android Jetpack*

ART

When an Android app is built within Android Studio it is compiled into an intermediate bytecode format (referred to as DEX format).

When the application is subsequently loaded onto the device, the Android Runtime (ART) uses a process referred to as Ahead-of-Time (AOT) compilation to translate the bytecode down to the native instructions required by the device processor.

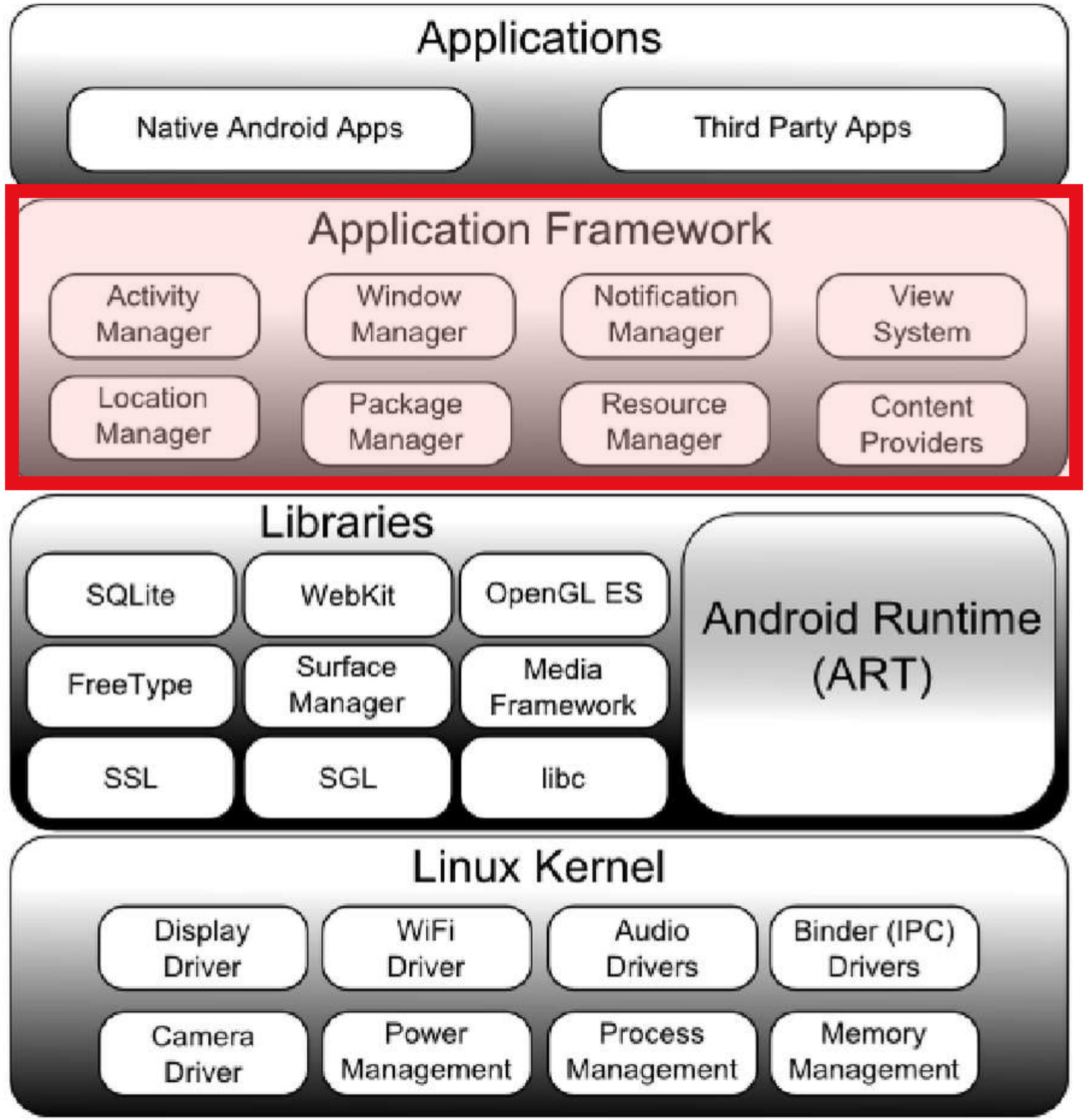


Smyth, Neil. *Android Studio 4. 0 Development Essentials - Java Edition : Developing Android Apps Using Android Studio 4. 0, Java and Android Jetpack*

Application Framework

The Application Framework is a set of services that collectively form the environment in which Android applications run and are managed.

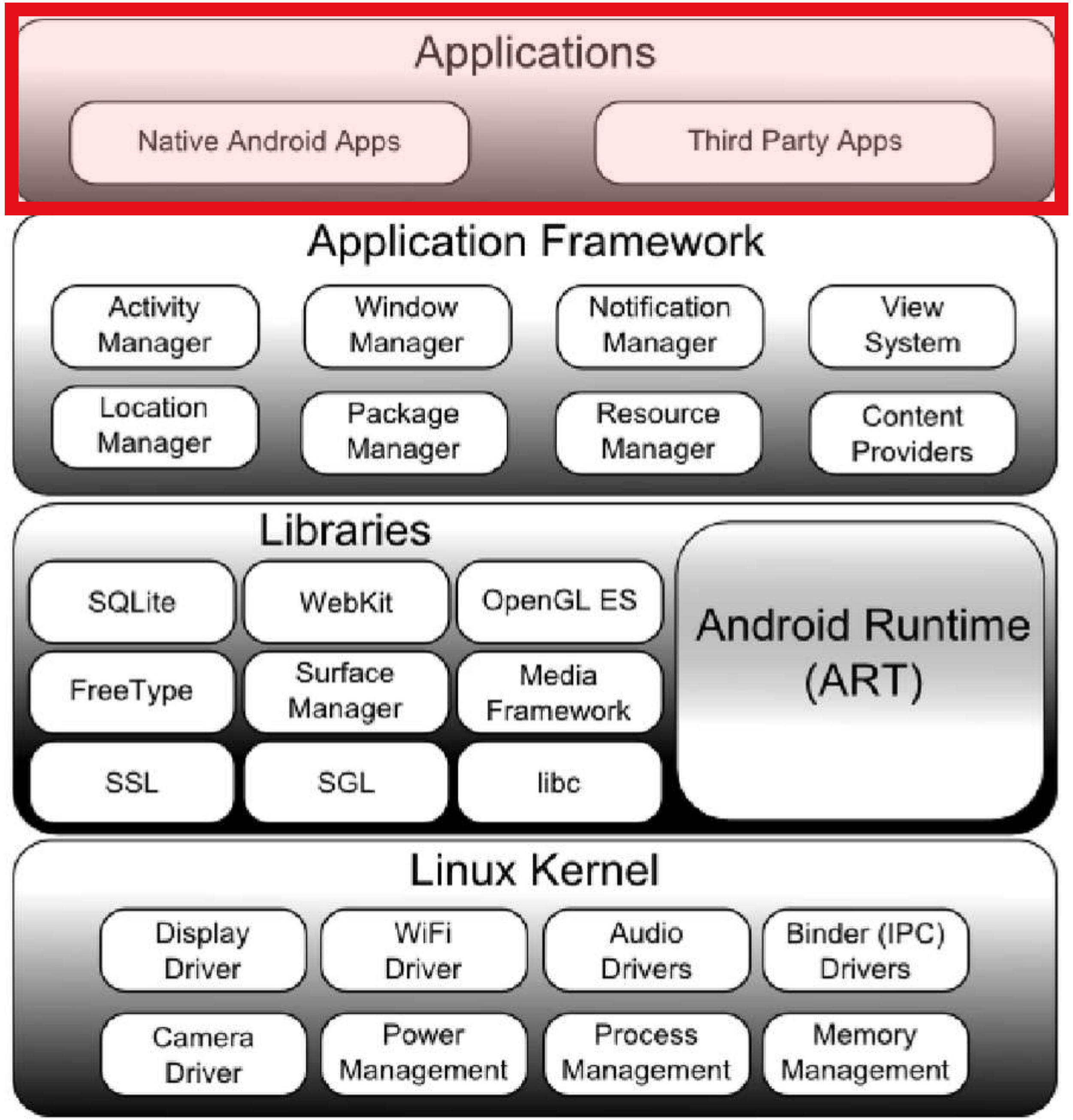
This framework implements the concept that Android applications are constructed from reusable, interchangeable and replaceable components.



Smyth, Neil. *Android Studio 4. 0 Development Essentials - Java Edition : Developing Android Apps Using Android Studio 4. 0, Java and Android Jetpack*

Applications

Located at the top of the Android software stack are the applications. These comprise both the native applications provided with the particular Android implementation (for example web browser and email applications) and the third party applications installed by the user after purchasing the device.



Smyth, Neil. *Android Studio 4. 0 Development Essentials - Java Edition : Developing Android Apps Using Android Studio 4. 0, Java and Android Jetpack*

Is Android code == Sun Java code?

Not exactly.

Android Java is not 100% Sun Java

If you need to get back in Java Coding:

https://www.youtube.com/watch?v=ksxHcuQA9sc&list=PLSzsOkUDsvdthwb-qGOrPl5wR73yqpdu3

…but they are very similar

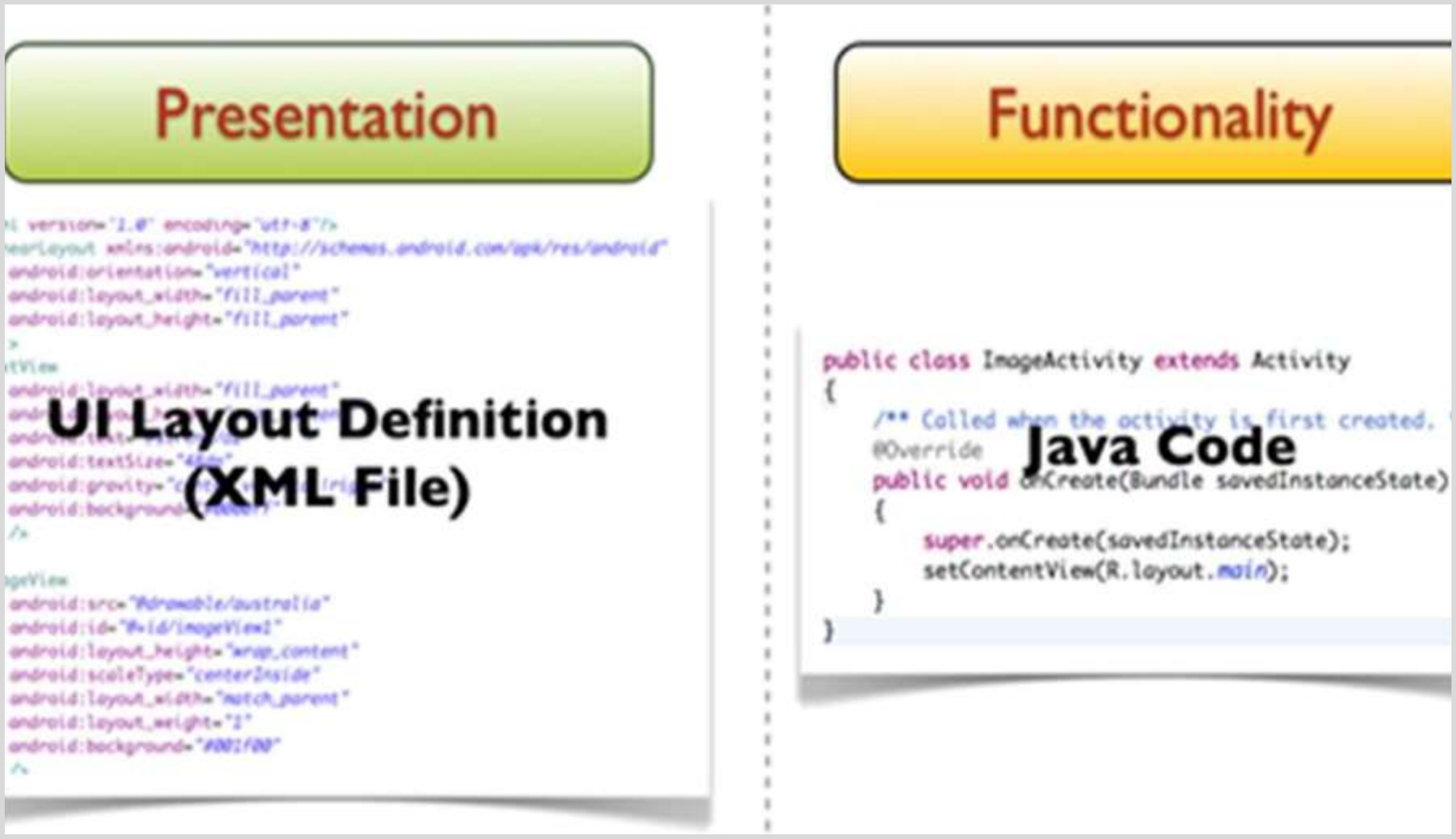
- Java is the language of Android. Large parts of Android are written in Java and its APIs are designed to be called primarily from Java.
- To develop Android apps, it is necessary to learn the concepts of Android, and then use them programmatically with Java.
- So the complexity of your Android apps will be limited by your knowledge of Java

- ***no shortcuts, you have to learn Java***

But wait …Kotlin?…..

https://youtu.be/czKo-jPVweg

The Android way : Separation



Presentation

UI Layout Definition (XML File)

```
l version="1.0" encoding="utf-8"?>
earLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:orientation="vertical"
 android:layout_width="fill_parent"
 android:layout_height="fill_parent"
 >
tView
 android:layout_width="fill_parent"
 andr[illegible]
 andr[illegible]
 android:textSize="[illegible]"
 android:gravity="[illegible]"
 android:background[illegible]
 />

geView
 android:src="@drawable/australia"
 android:id="@+id/imageView1"
 android:layout_height="wrap_content"
 android:scaleType="centerInside"
 android:layout_width="match_parent"
 android:layout_weight="1"
 android:background="#001f00"
 />
```

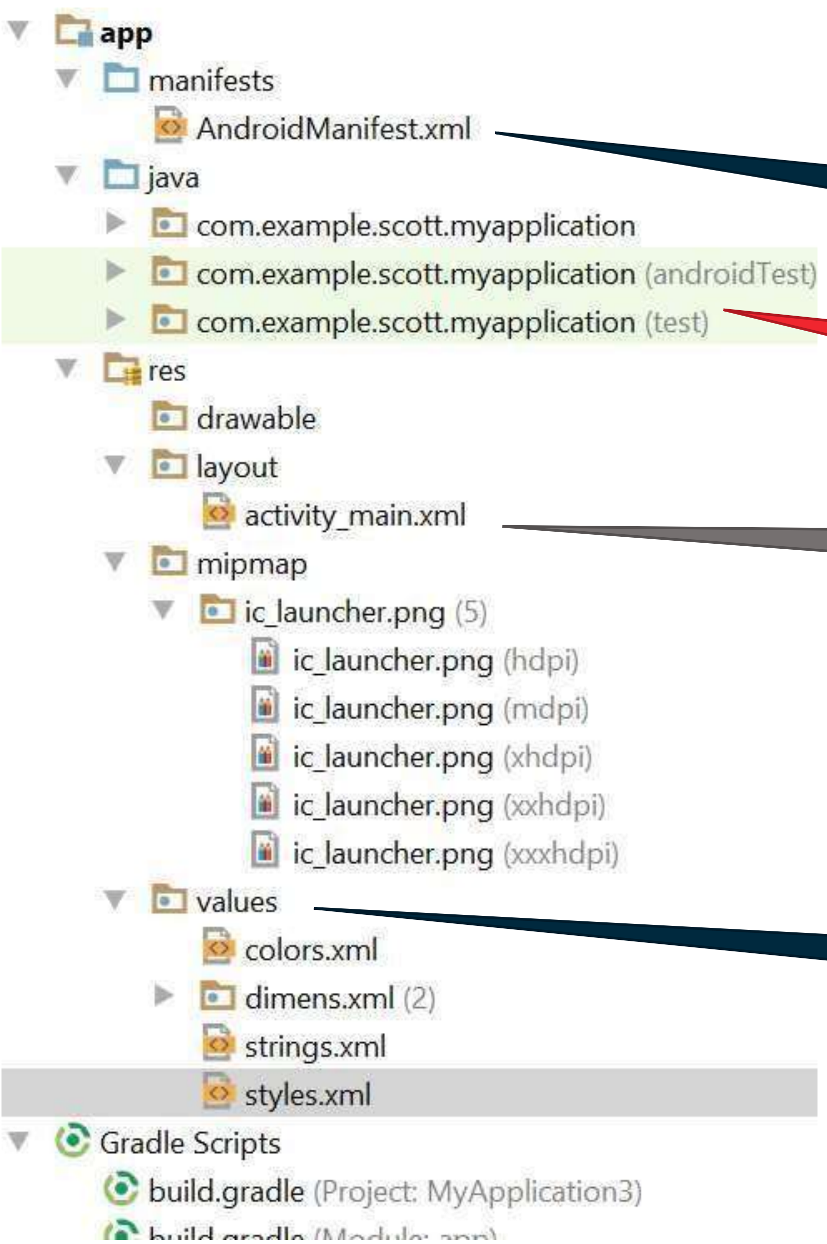
Functionality

Java Code

```
public class ImageActivity extends Activity
{
    /** Called when the activity is first created. 
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Image: http://apcmag.com/building-a-simple-android-app.htm/

Anatomy of an Android Application



```
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▶ com.example.scott.myapplication
    ▶ com.example.scott.myapplication (androidTest)
    ▶ com.example.scott.myapplication (test)
  ▼ res
      drawable
    ▼ layout
        activity_main.xml
    ▼ mipmap
      ▼ ic_launcher.png (5)
          ic_launcher.png (hdpi)
          ic_launcher.png (mdpi)
          ic_launcher.png (xhdpi)
          ic_launcher.png (xxhdpi)
          ic_launcher.png (xxxhdpi)
    ▼ values
        colors.xml
      ▶ dimens.xml (2)
        strings.xml
        styles.xml
▼ Gradle Scripts
    build.gradle (Project: MyApplication3)
    build.gradle (Module: app)
```

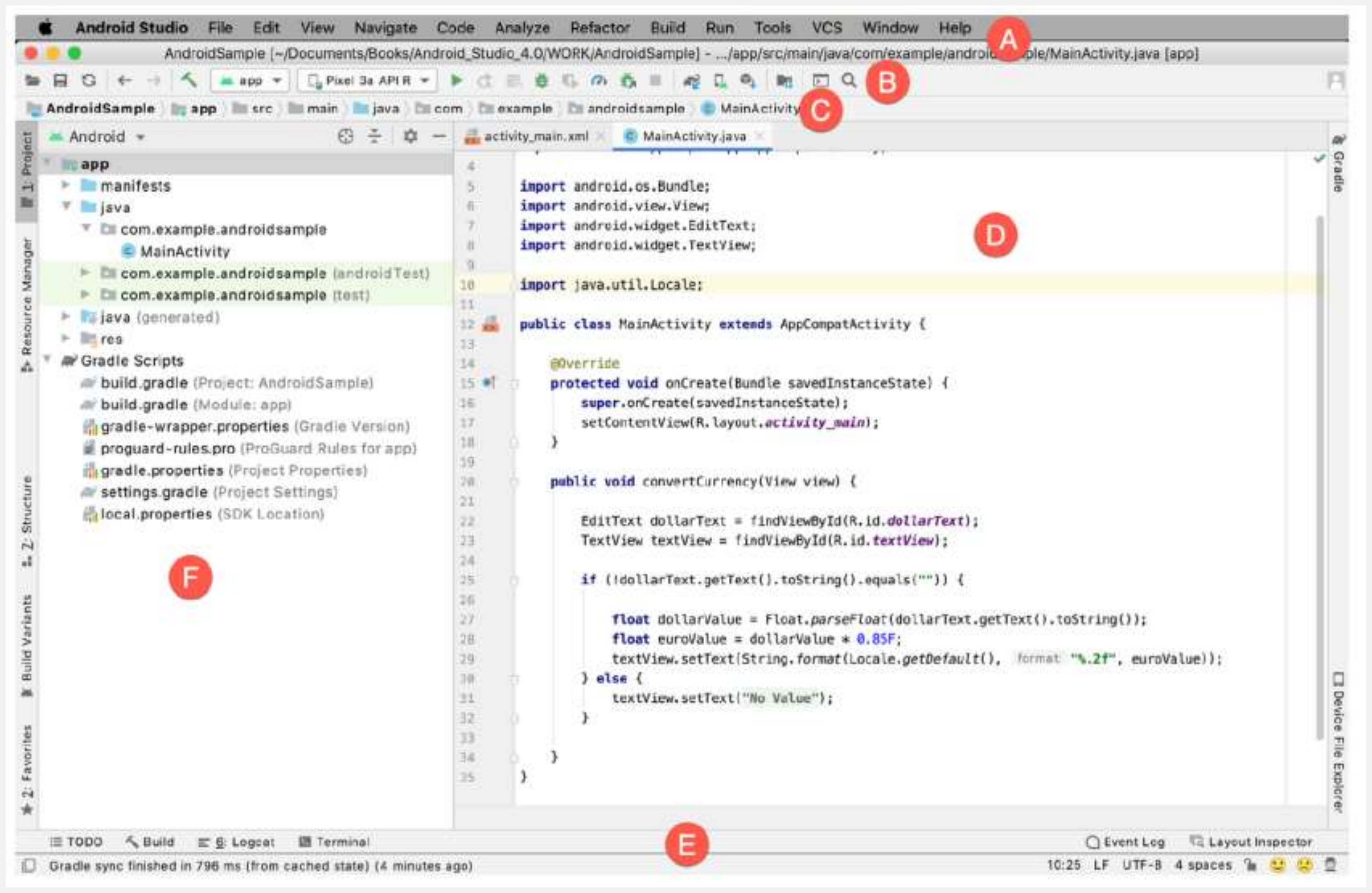
App configuration (XML) to OS & VM

App package, where your activity file lives (JAVA)

Activity layout (XML)

Values used in your app

Getting Started with Android Studio


Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
A
AndroidSample [~/Documents/Books/Android_Studio_4.0/WORK/AndroidSample] - .../app/src/main/java/com/example/androi...ple/MainActivity.java [app]
app
Pixel 3a API R
B
AndroidSample app src main java com example androidsample MainActivity
C
Android
activity_main.xml
MainActivity.java
app
manifests
java
com.example.androidsample
MainActivity
com.example.androidsample (androidTest)
com.example.androidsample (test)
java (generated)
res
Gradle Scripts
build.gradle (Project: AndroidSample)
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)
F
D
1: Project
Resource Manager
Z: Structure
Build Variants
2: Favorites
Gradle
Device File Explorer
TODO
Build
6: Logcat
Terminal
E
Event Log
Layout Inspector
10:25 LF UTF-8 4 spaces

```java
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

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

    public void convertCurrency(View view) {

        EditText dollarText = findViewById(R.id.dollarText);
        TextView textView = findViewById(R.id.textView);

        if (!dollarText.getText().toString().equals("")) {

            float dollarValue = Float.parseFloat(dollarText.getText().toString());
            float euroValue = dollarValue * 0.85F;
            textView.setText(String.format(Locale.getDefault(), format "%.2f", euroValue));
        } else {
            textView.setText("No Value");
        }

    }
}
```

Gradle sync finished in 796 ms (from cached state) (4 minutes ago)

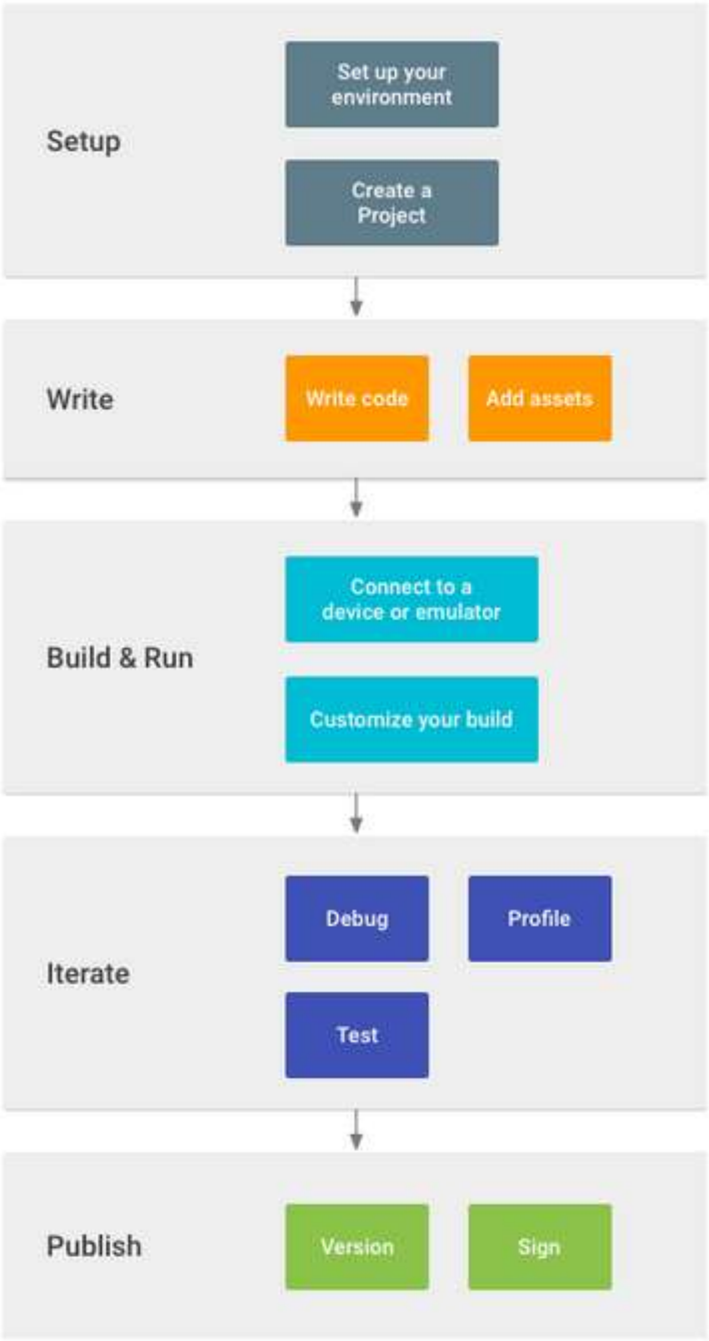
A. Menu Bar
B. Toolbar
C. Navigation Bar
D. Editor Window
E. Status Bar

Android Studio comes with…

- IntelliJ IDE + Android Studio plugin
- Android SDK Tools
- Android Platform-tools
- A version of the Android platform
- Android Emulator with an Android system image including Google Play Services

…More on these in next week's labs

Developing an Android App: The Basic Steps

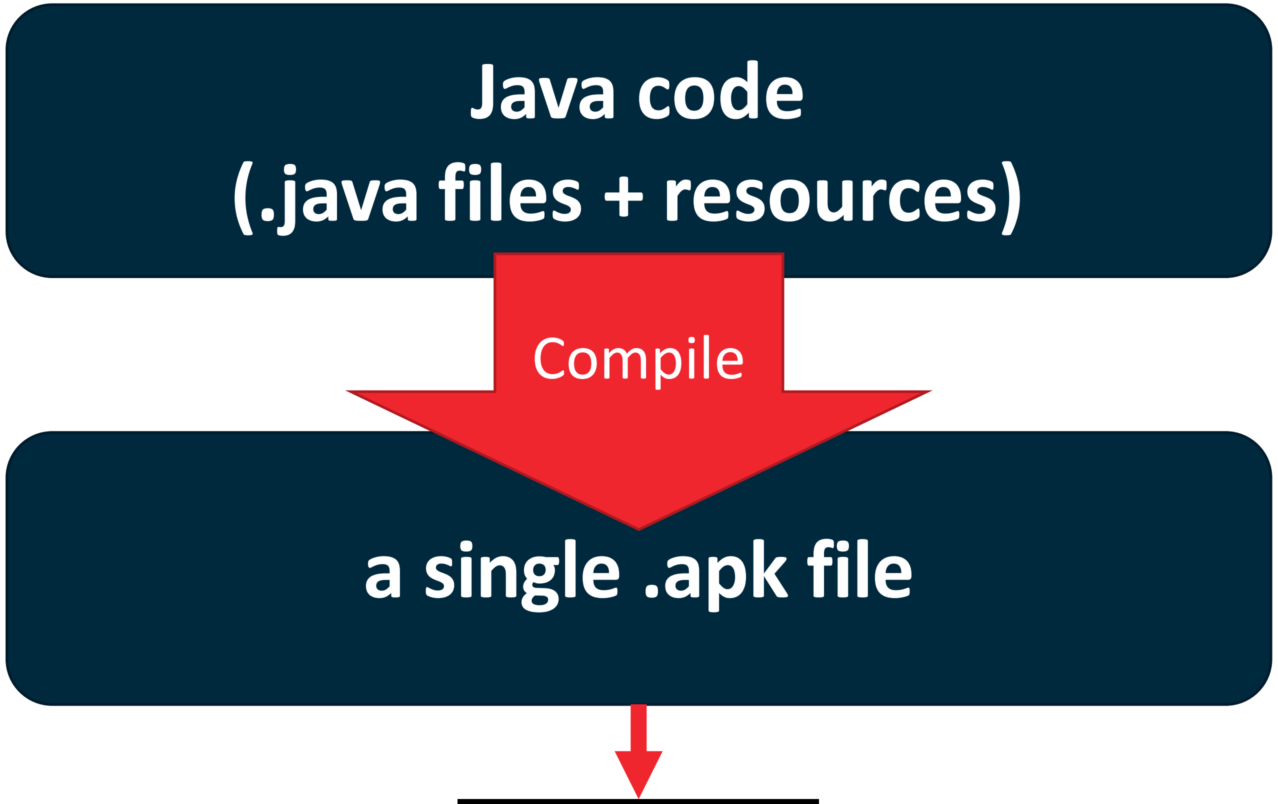


The workflow to develop an app for Android is conceptually the same as other app platforms.

For more information visit the link below.

https://developer.android.com/studio/workflow

Applications Fundamentals



Java code
(.java files + resources)

Compile

a single .apk file

https://developer.android.com/guide/components/fundamentals.html

Android Applications – key ideas

- **Component based software development:**
  - builds on top of object-oriented programming concepts (e.g., an application does not just comprise of any user-defined classes but at least must have particular prescribed classes plus one or more user-defined classes)
  - files? code + data (used in code)
  - components (objects) have their own lifecycles
- **Event-driven programming**
  - event: what happens with objects
  - listeners: attached to events in order to listen for events and program methods respond to them (“on…(…)”)
- **Multi-threading (see Thread class)**
  - parallel execution
  - create new Threads, start them

Structure of an Android application

An Android application is composed of one or more application components (activities, services, content providers, and broadcast receivers) that talk to each other (exchange messages called intents)

Components & Description
Activities They dictate the UI and handle the user interaction to the smart phone screen.
Services They handle background processing associated with an application.
Broadcast Receivers They handle communication between Android OS and applications.
Content Providers They handle data and database management issues.

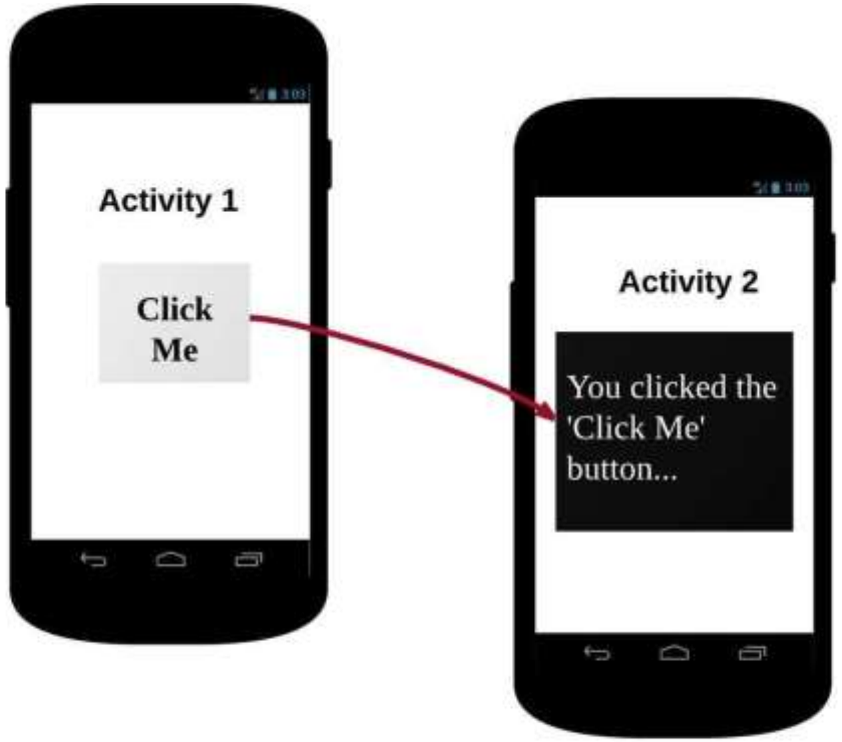
https://www.tutorialspoint.com/android/android_application_components.htm

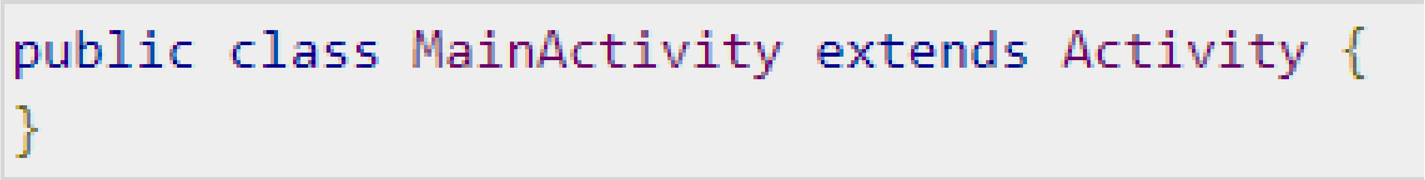
Android Components - Activity

Activity: An activity represents a single screen with a user interface.

- One app can have many activities.
- Typically, each Android app has one activity that is specified as the "main" activity, which is presented to the user when launching the app.

Credit: Chuong Vo for CSE4MPC





```
public class MainActivity extends Activity {
}
```

https://www.tutorialspoint.com/android/android_application_components.htm

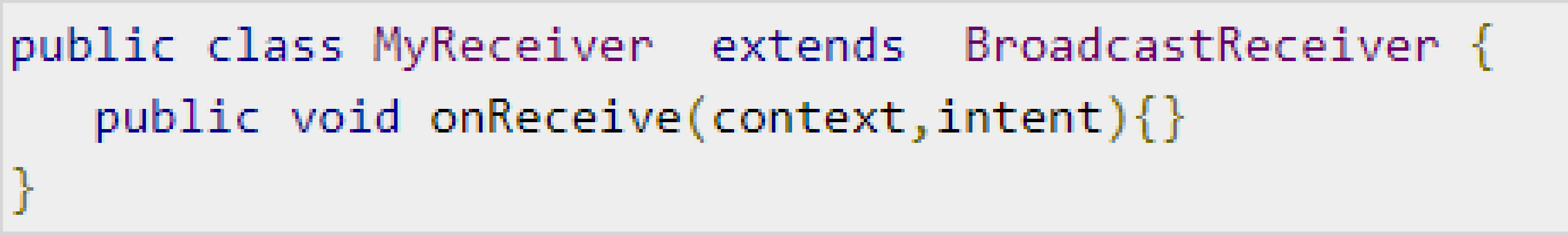
Android Components – Broadcast Receiver

Broadcast Receiver: Used to respond to system-wide broadcast announcements (events).

- has no user interface.
- Can be used as a “gateway” to trigger other components.

Credit: Chuong Vo forCSE4MPC

A broadcast receiver is implemented as a subclass of **BroadcastReceiver** class and each message is broadcaster as an **Intent** object.



```
public class MyReceiver  extends  BroadcastReceiver {
   public void onReceive(context,intent){}
}
```

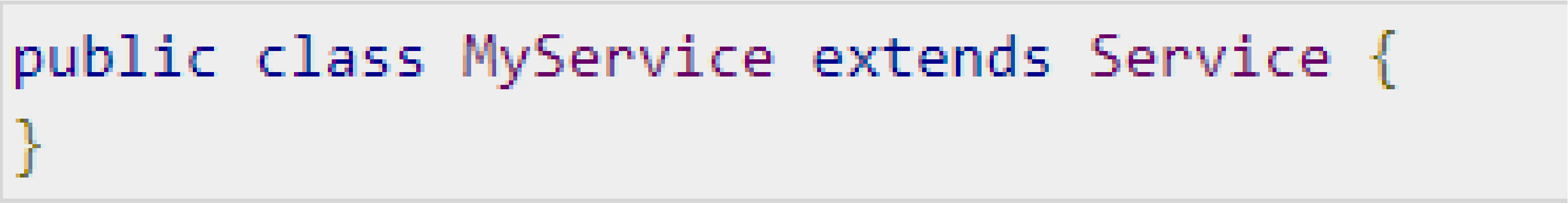
https://www.tutorialspoint.com/android/android_application_components.htm

Android Components - Service

Service: Used to perform long-running operations in background.

- Has no user interface.
- One app can have many services.
- A service can be automatically triggered to start by a system-wide event (e.g., device-boot completed event)

Credit: Chuong Vo forCSE4MPC



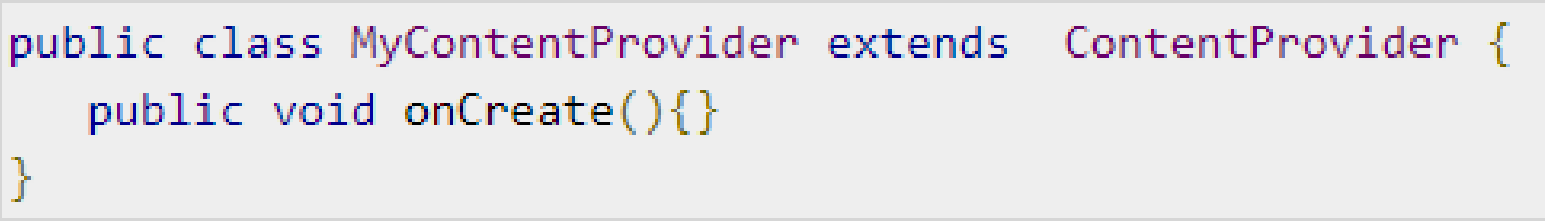
```java
public class MyService extends Service {
}
```

https://www.tutorialspoint.com/android/android_application_components.htm

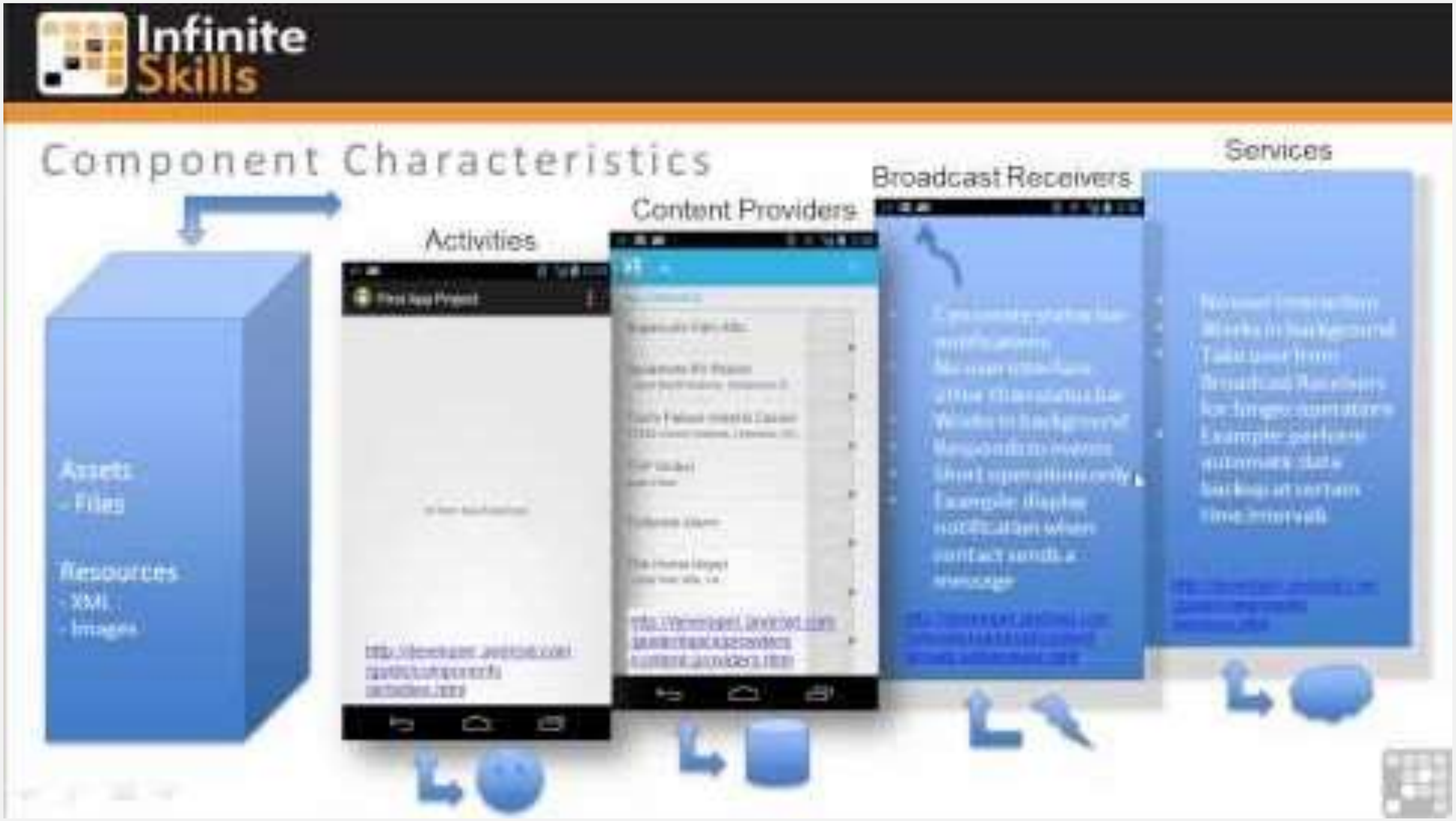
Android Components - Content Provider

Content Provider: Used to manage the data of the app.

- You can store the data in the file system, or an SQLite database...
- The data can be shared between apps via the content providers.



```
public class MyContentProvider extends  ContentProvider {
   public void onCreate(){}
}
```



Infinite Skills

Component Characteristics

Assets
- Files

Resources
- XML
- Images

Activities

http://developer.android.com/guide/components/activities.html

Content Providers

http://developer.android.com/guide/topics/providers/content-providers.html

Broadcast Receivers

- Can create status bar notifications
- No user interface (other than status bar)
- Works in background
- Responds to events
- Short operations only
- Example: display notification when contact sends a message

[illegible]

Services

- No user interaction
- Works in background
- Takes over from Broadcast Receivers for longer operations
- Example: perform automatic data backup at certain time intervals

[illegible]

https://youtu.be/C9W4WmokJFk?t=3

Components Example

A prime example is a media player playing songs from a play list.

- The player application would probably have **one or more activities** that allow the user to choose songs and start playing them.
- However, the music playback itself would not be handled by an activity because users will expect the music to keep playing even after they leave the player and begin something different.
- To keep the music going, the media player activity could start a **service** to run in the background.
- The system would then keep the music playback service running even after the activity that started it leaves the screen.

https://play.google.com/store/apps/details?id=com.musicplayer.playermusic

Starting components

Content providers are activated when they're targeted by a request from a ContentResolver. The other three components — activities, services, and broadcast receivers — are activated by asynchronous messages called intents.

- An intent may convey some data which the component being started might need.

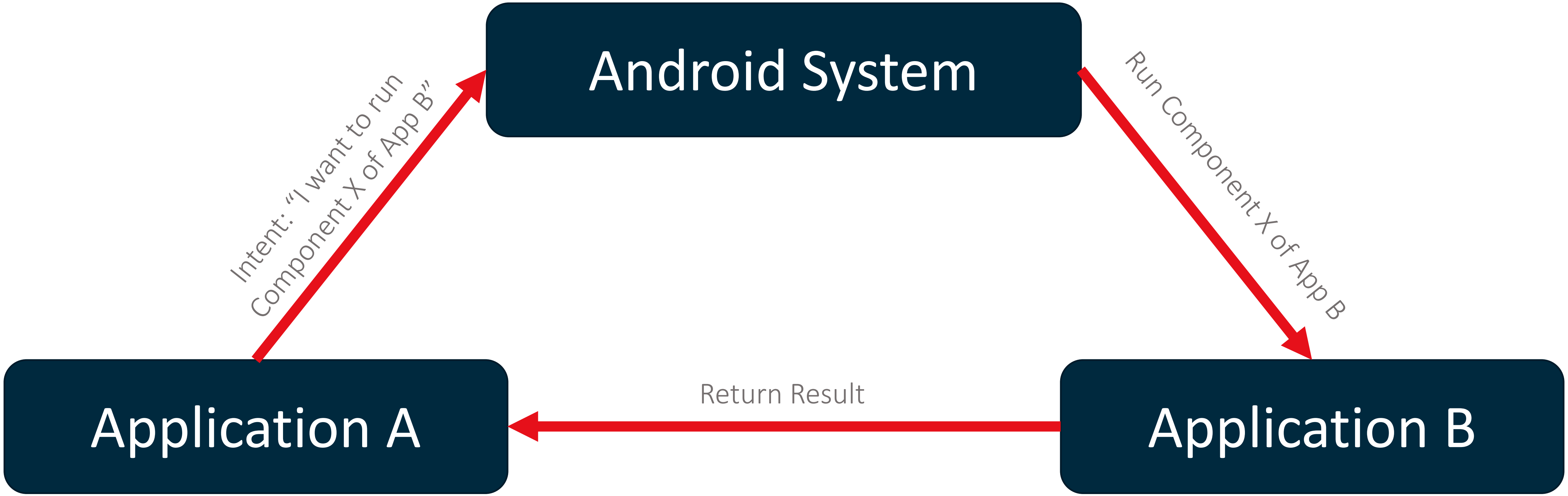
  For example, it might convey a request for an activity to present an image to the user or let the user edit some text. For broadcast receivers, the Intent object names the action being announced. For example, it might announce to interested parties that the camera button has been pressed."

- A component can start an activity and receive a result when that activity finishes.

https://developer.android.com/guide/components/fundamentals

Starting components con’t

- A **unique** aspect of Android is that any app can startanother app’s component.



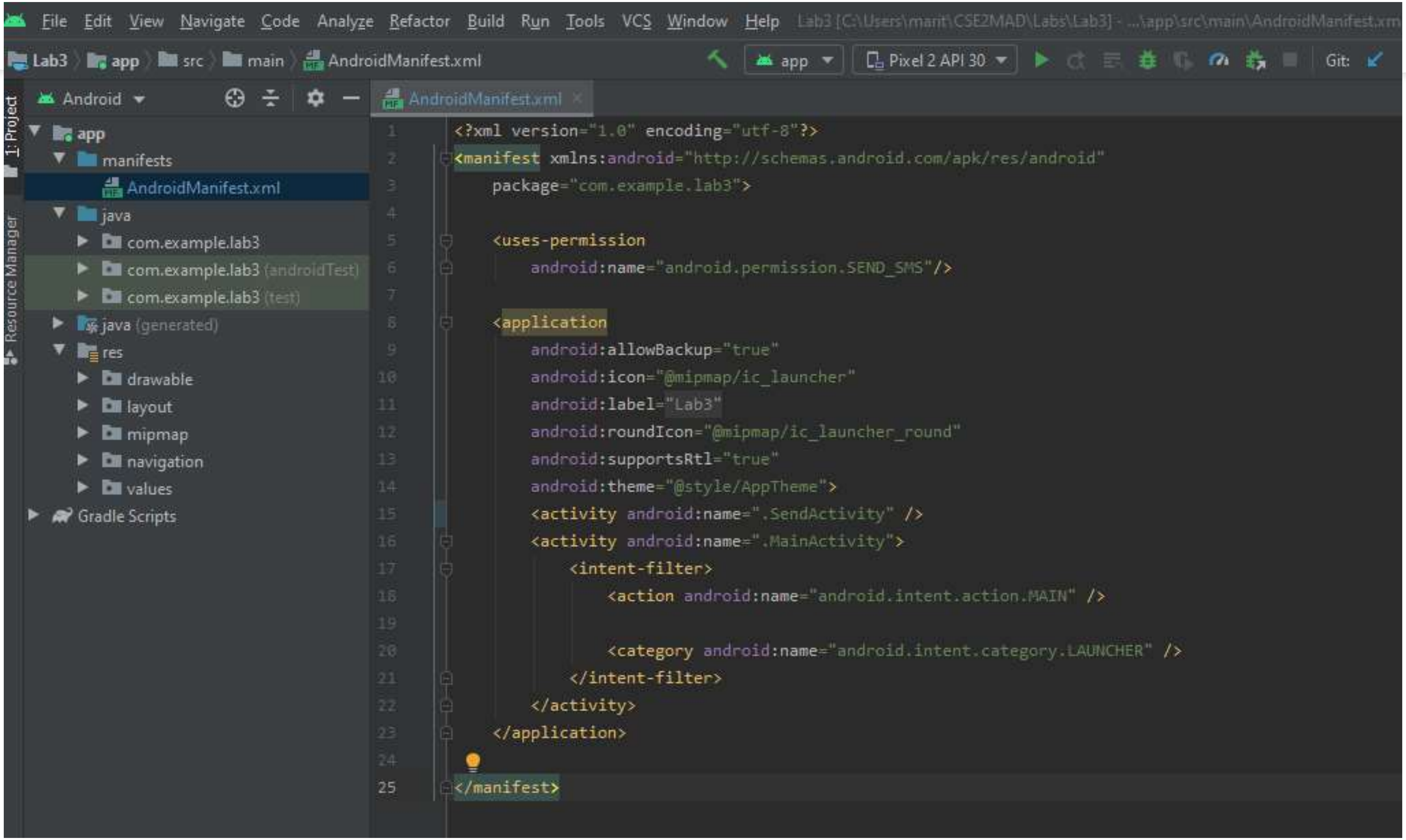
The app manifest file

To let the system knows about your app the app's AndroidManifest.xml declares

- all components of the app,
- any required user-permissions (e.g., GPS, Internet…)
- the minimum API level required by the app
- hardware and software features required by the app (e.g., camera, blue-tooth…)
- API libraries (other than the core APIs), such as the Google Maps APIs

Credit: Prepared by Chuong Vo forCSE4MPC

The app manifest file



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Lab3 [C:\Users\marit\CSE2MAD\Labs\Lab3] - ...\app\src\main\AndroidManifest.xm
Lab3 app src main AndroidManifest.xml
app
Pixel 2 API 30
Git:
1: Project
Resource Manager
Android
app
manifests
AndroidManifest.xml
java
com.example.lab3
com.example.lab3 (androidTest)
com.example.lab3 (test)
java (generated)
res
drawable
layout
mipmap
navigation
values
Gradle Scripts
AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lab3">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Lab3"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".SendActivity" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Application resources

- Images Audio files
- XML files define **menus**, **styles**, **colors**, **strings**, **animations**, and the **layout** of activity user interfaces.
- The SDK tool automatically assigns **a unique integer ID for every resource** included in your Android project, which you can use to **reference** the resource from your application code or from other resources defined in XML files.

Credit: Prepared by Chuong Vo forCSE4MPC

Summary

- Android architecture
- Separation of view & functionality
- Android Application components