ANDROID OVERVIEW AND BASIC

Nam Nguyen Tu

CSC Proprietary and Confidential

Introduction

- Your role
- Your background and experience in the subject
- What do you want from this course

Course Objectives

- At the end of the course, you will have acquired sufficient knowledge to:
- Understand Android and it’s definitions
- Understand how to make an Android application
- Create a simple application on Mobile

Agenda

Course Audience and Prerequisite

- The course is for programmers which interest in Android development.
- The following are prerequisites to this course:
  - Java and OO knowledge
  - XML knowledge
  - MVC/MVP or MVVM knowledge

Assessment Disciplines

- Class Participation: 40%
- Assignment: 60%
- Final Exam: 0%
- Passing Scores: 70%

Set Up Environment

- To complete the course, your PC must install:
  - Android SDK
  - Eclipse with Android plugins

Course Administration

- In order to complete the course you must:
  - Sign in the Class Attendance List
  - Participate in the course
  - Provide your feedback in the End of Course Evaluation

ANDROID INTRODUCTION

Introduction about Android and its architecture

ANDROID INTRODUCTION

- What is Android ?

- The world's most popular mobile platform which can be run on mobiles and tablets.

- There are more than 600,000 apps and games available on Google Play

- Developed by the Open Handset Alliance led by Google

ANDROID INTRODUCTION

- Main hardware manufacturers:

- Main versions are:





ANDROID INTRODUCTION

- Games

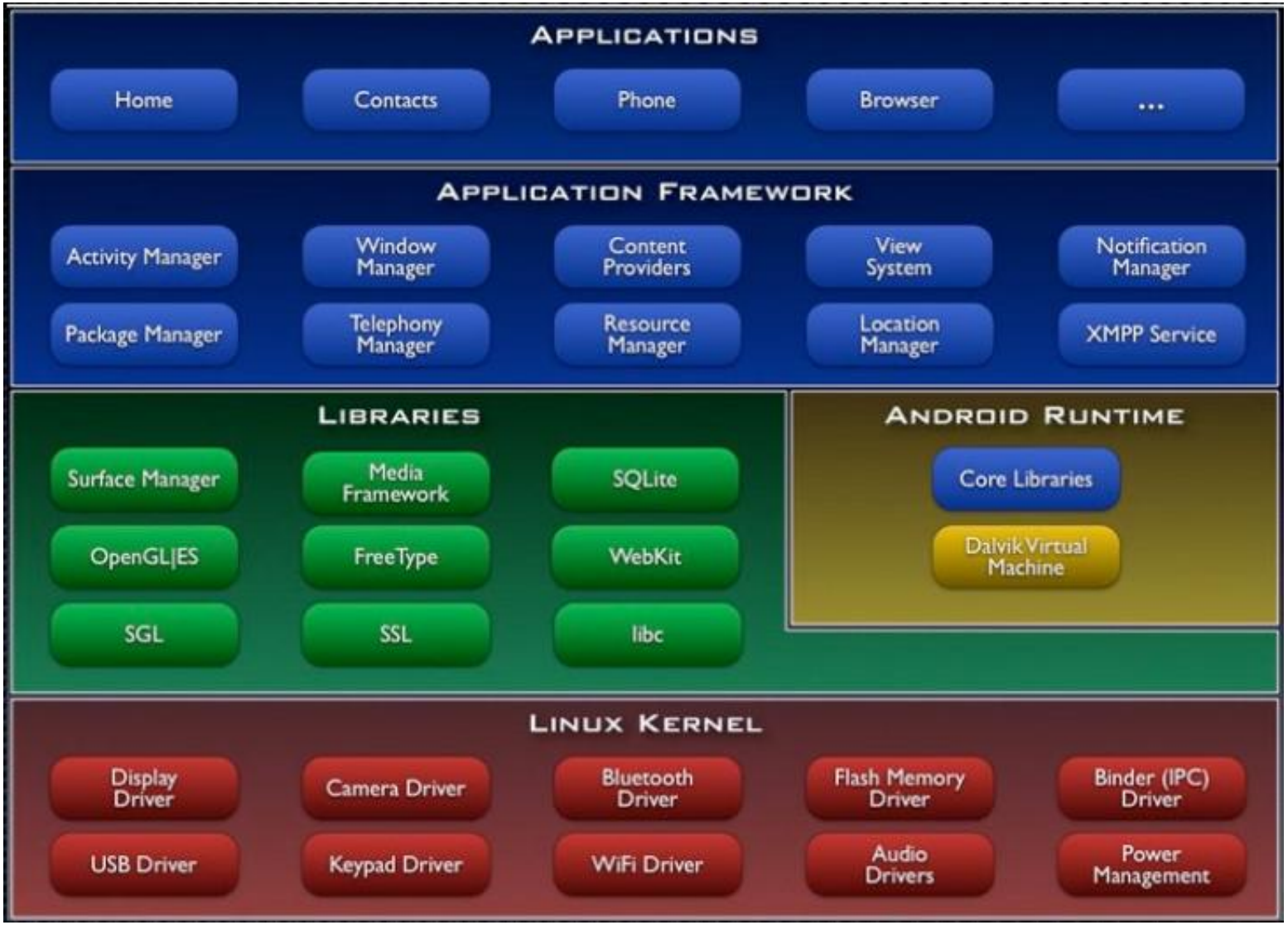
- Applications

So, what is Android architecture ?

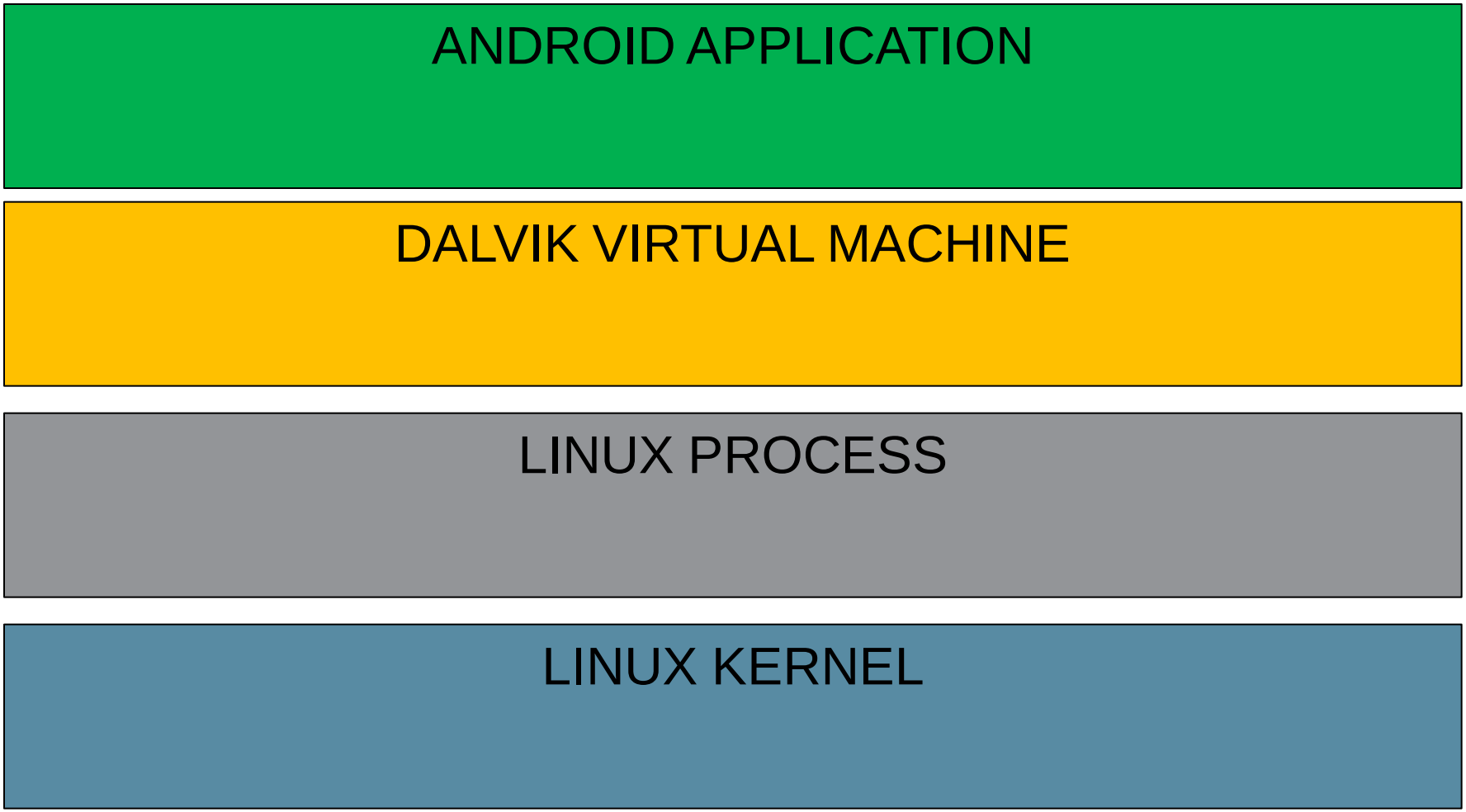
What is behind Android ?

What is make Android so helpful ?

ANDROID INTRODUCTION

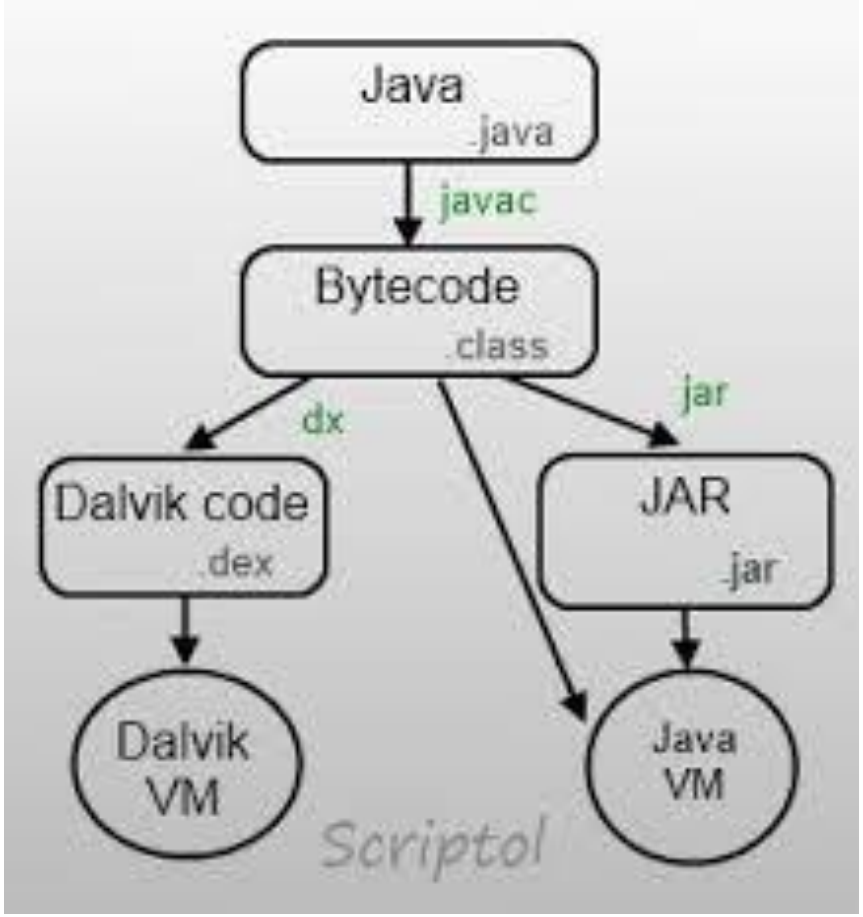


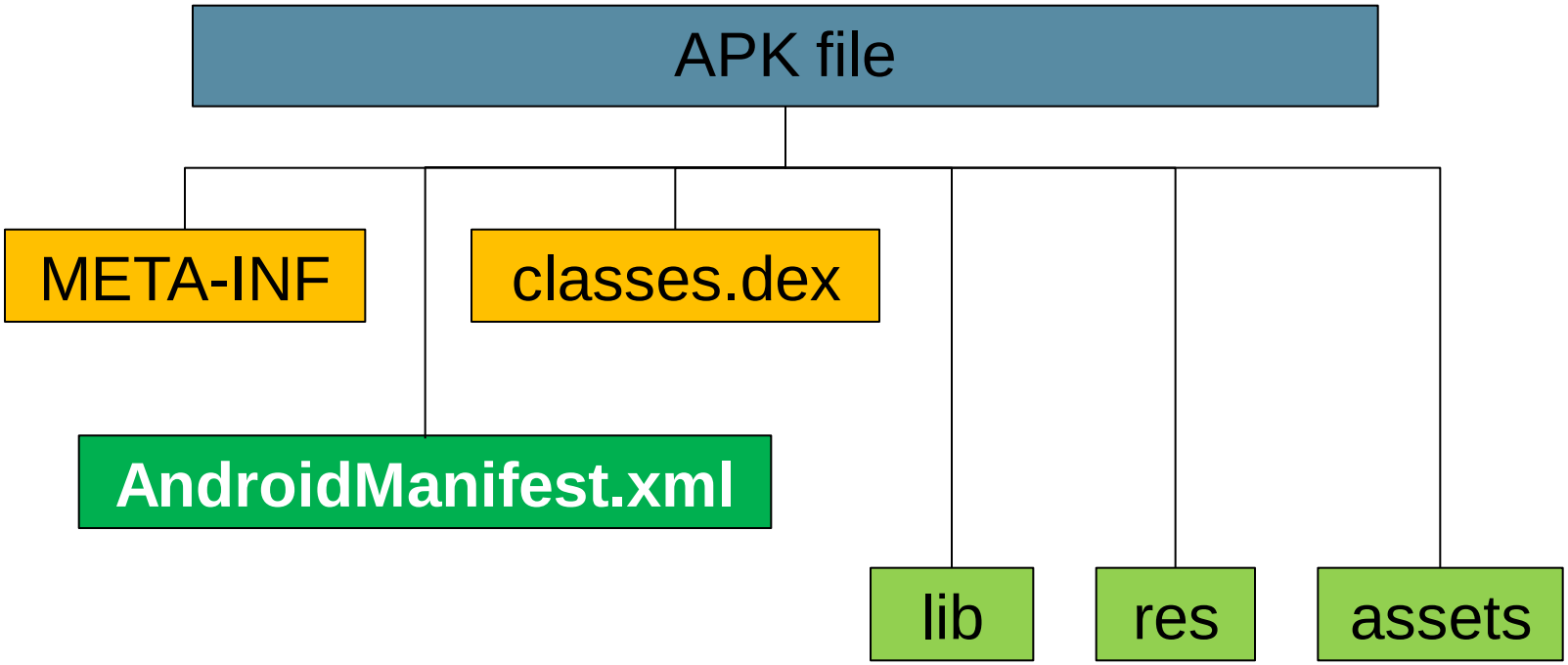
ANDROID INTRODUCTION



ANDROID INTRODUCTION

- Dalvik Virtual Machine
  - A register-based architecture
  - Convert Java classes to .dex format to execute on Dalvik.
  - VM was slimmed down
  - Use 16-bit instruction set instead of 8-bit of Java
  - JIT
  - Optimized garbage collection




APK file
META-INF
classes.dex
AndroidManifest.xml
lib
res
assets

ANDROID DEVELOPMENT

Create Android application

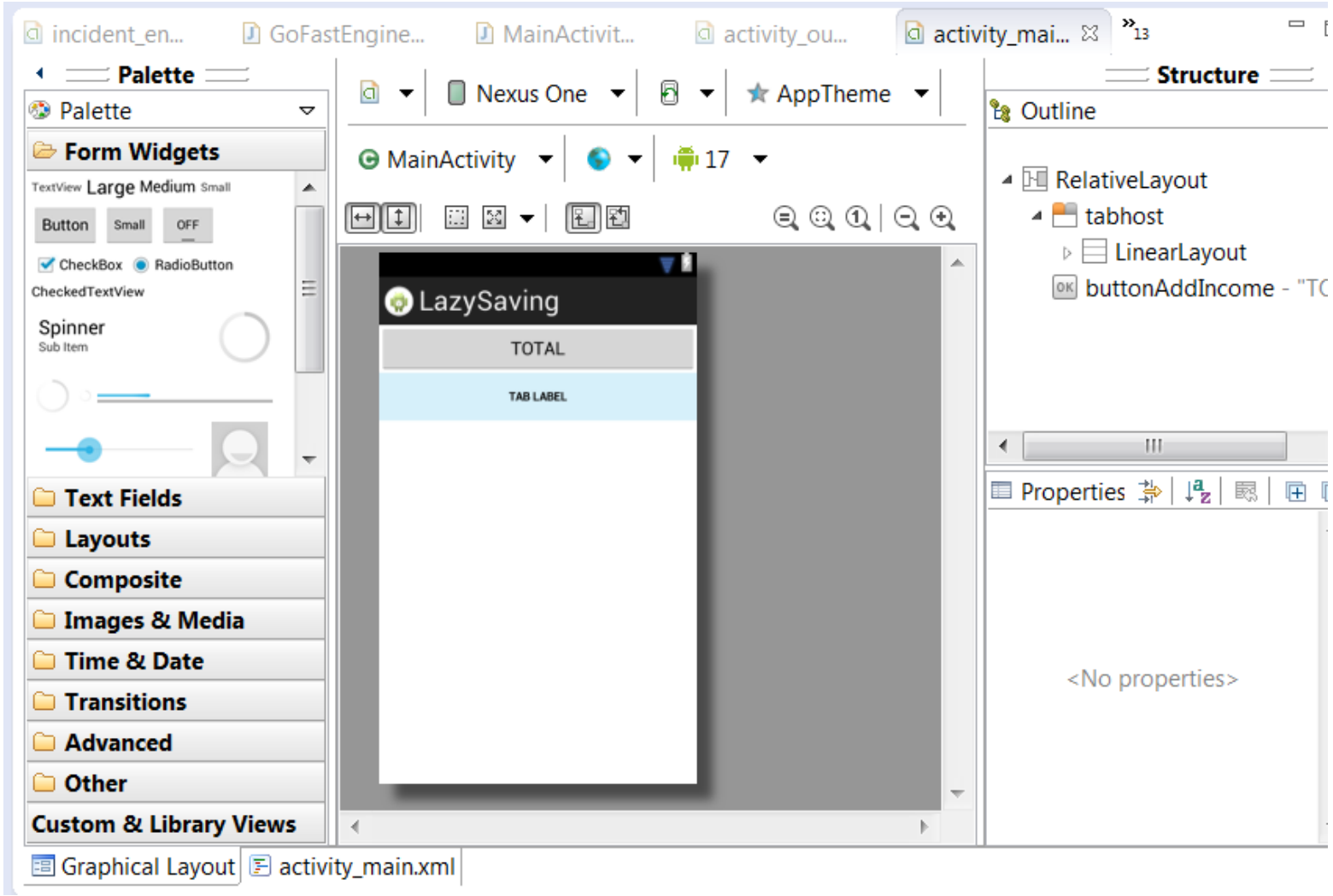
ANDROID DEVELOPMENT TOOLS

- **Java**
  - Google Android SDK
  - Google Android NDK
- **Basic**
  - Basic4Android
- **C#**
  - Xamarin Android
  - Dot42
  - Game development tool: Unity
- **Ruby, Python, Lua**
  - Script Layer for Android
  - Game development tool such as Corona, Marmalade etc ..

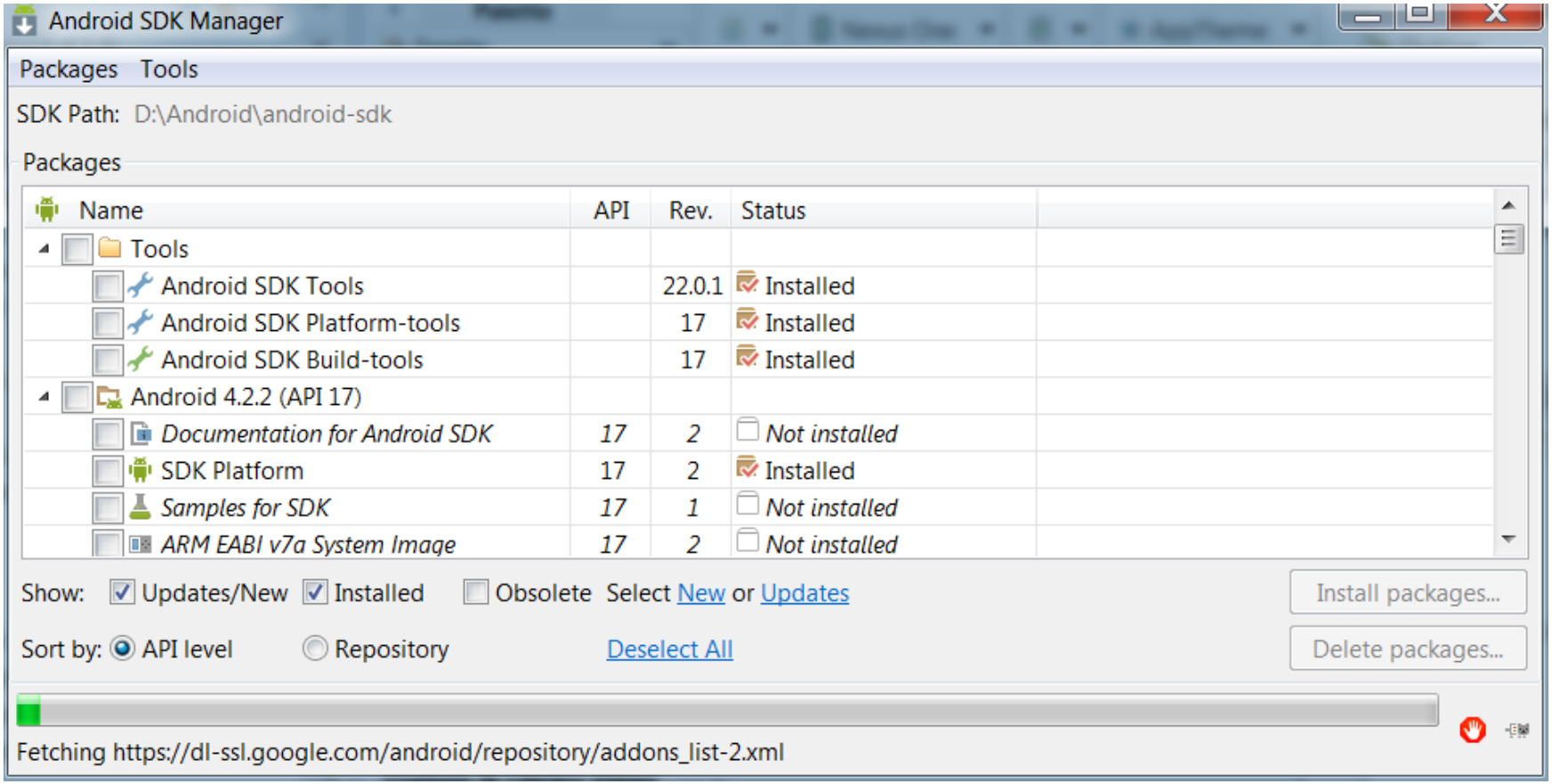
ANDROID DEVELOPMENT TOOLS

- We put focus on Java and Google SDK !
- It is mainstream tool and technique.
- It has GUI Builder and IDE
- It is supported by Google.

ANDROID DEVELOPMENT TOOLS



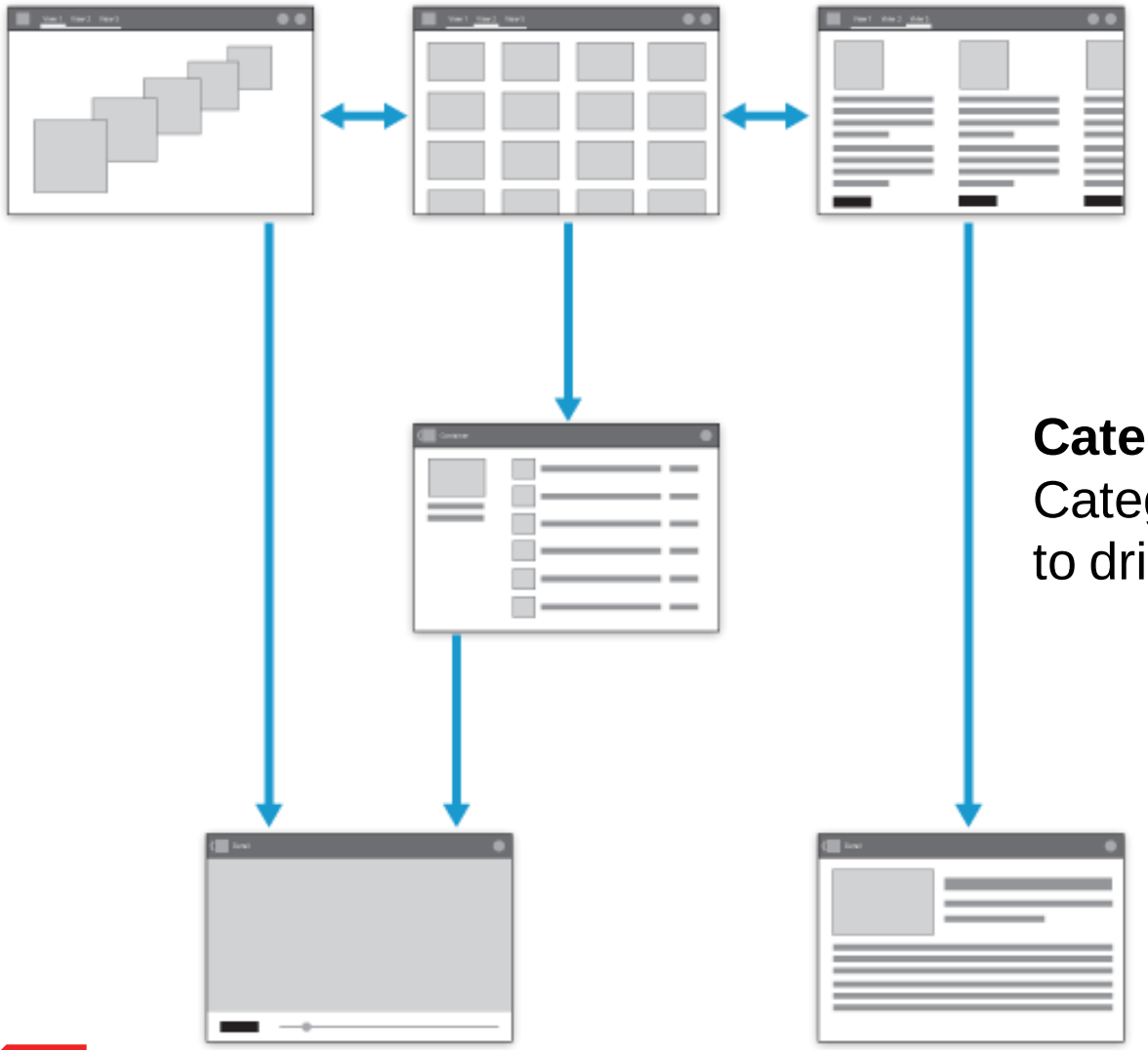
ANDROID DEVELOPMENT TOOLS



HOW TO DEVELOP AN APPLICATION

- **Idea**
- **Concept it by storyboard**
- **Prototype**
- **Coding**
- **Test on emulator**
- **Test on real device**
- **Sign application**
- **Give Google 25$ for one time only**
- **Upload application to Google Play.**

APPLICATION STRUCTURE



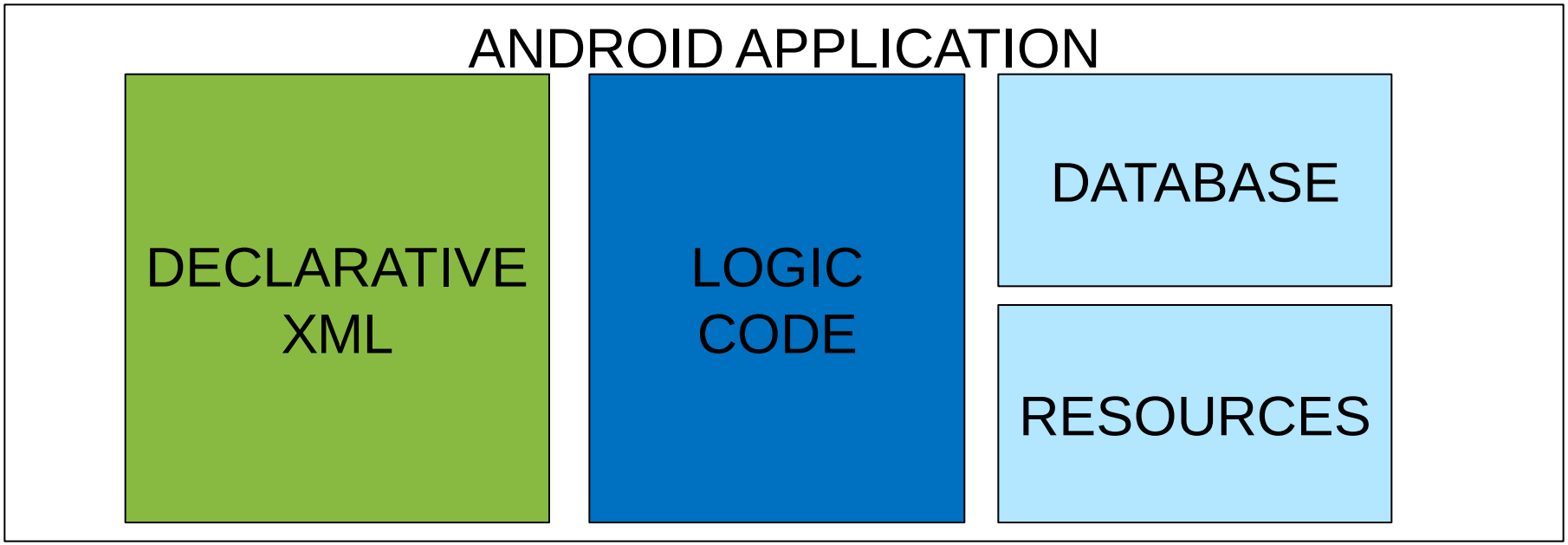
Top level views
The top level of the app typically consists of the different views that your app supports.

Category views
Category views allow you to drill deeper into your data.

Detail/edit view
The detail/edit view is where you consume or create data.

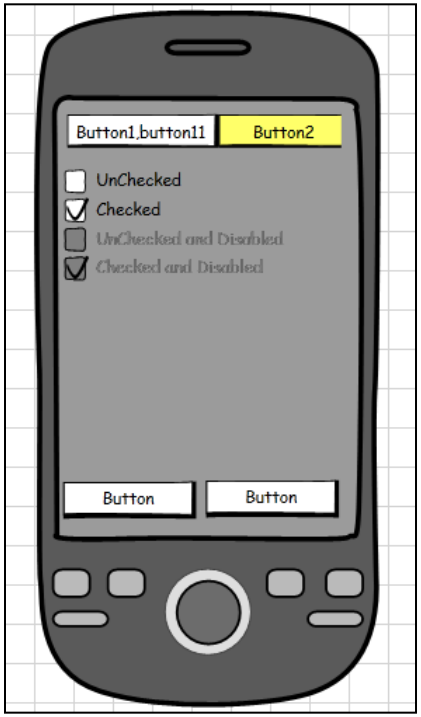
HOW TO CODE ?

- Declarative UI in XML
  - to describe how something looks or appears such what a button should look like.
- Programming Logic
  - Control the flow of screen and business logic inside application



DECLARE UI IN XML ??

- Describe User Interface in XML



activity_intro.xml

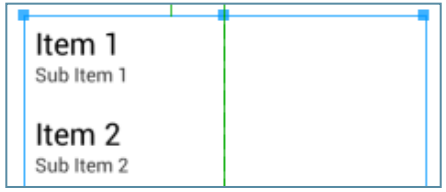
```
<RelativeLayout xmlns:android="http://schemas.andr
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertica
    android:paddingLeft="@dimen/activity_horizonta
    android:paddingRight="@dimen/activity_horizont
    android:paddingTop="@dimen/activity_vertical_m
    tools:context=".IntroActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

It has WYSWYG tool !!!!

- **VIEW ?**
- Everything you see on screen is view.
  - TextView (text)

  - ButtonView (buttons)

  - ListView (lists)



  - EditView (text boxes)

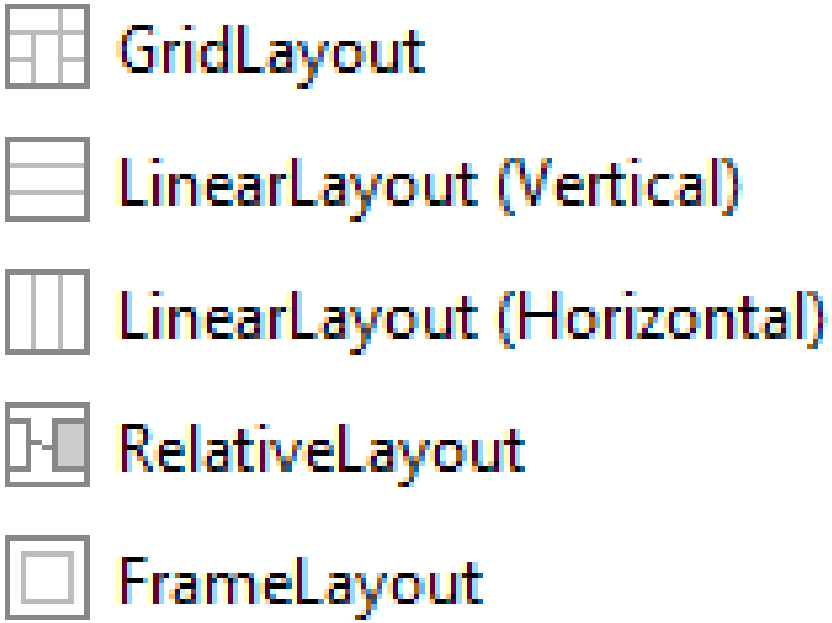
•VIEW ?

- Everything you see on screen is view.
  - ImageView

  - DatePicker

  - WebView
    (embedded web + webkit rendering)

- **LAYOUT ?**
- Organize and group views together.



VIEW AND LAYOUT

- **LAYOUT ?**
- Resizable
- Auto-expand with the different sizes in different devices
- Customizable

What is CODE ?

CODE BEHIND

- To make UI elements respond to user interactions
- To inter exchange data between UI elements
- To perform application logic
- To run in background and perform background works
- To do the work when saving data in store
- Stays in application components.

APPLICATION COMPONENTS

- Activity
- Intent
- Service
- Content Provider
- Broadcast Receiver

ACTIVITY

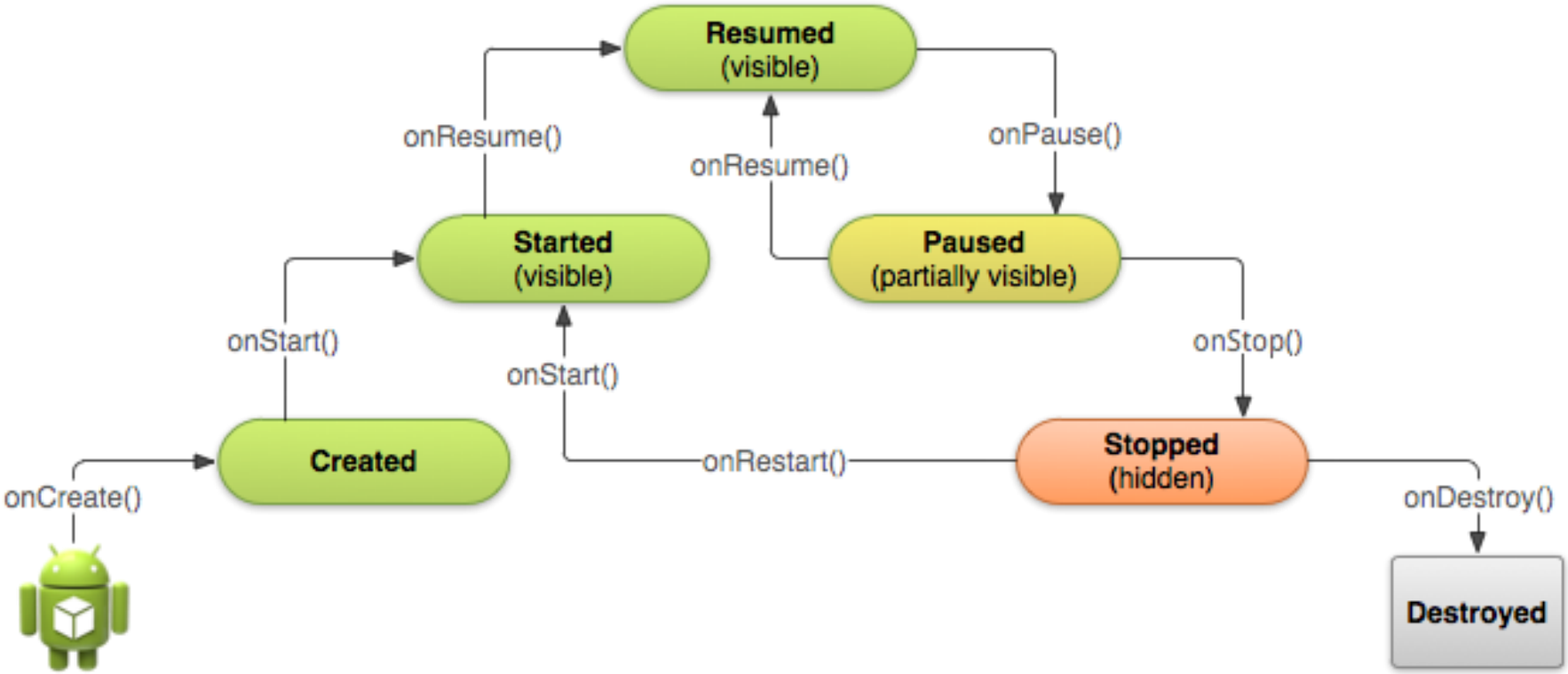
- Stay behind and control the View
- Listen and respond to events from View.

```
public class SomeClass extends Activity {
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);}

public void handlerMethod(View clickedButton) {
String someName = getString(R.string.some_name);
doSomethingWith(someName);
} }
```

res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout …>
<TextView … />
<Button … android:onClick="handlerMethod" />
</LinearLayout>
```

ACTIVITY LIFECYCLE

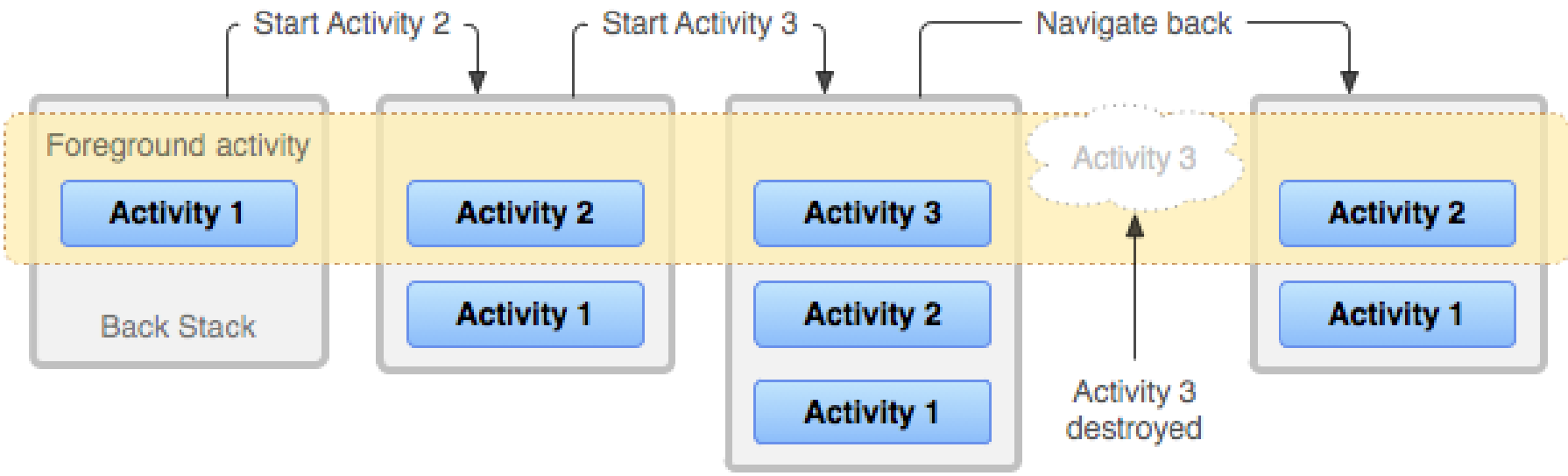


ACTIVITY LIFECYCLE

- Resumed
  - Activity is in the foreground and the user can interact with it.
- Paused
  - Activity is partially obscured by another activity—the other activity that's in the foreground is semi-transparent or doesn't cover the entire screen.
  - The paused activity does not receive user input and cannot execute any code.
- Stopped
  - Activity is completely hidden and not visible to the user;
  - In the background.
  - Activity instance and all its state information such as member variables is retained, but it cannot execute any code.

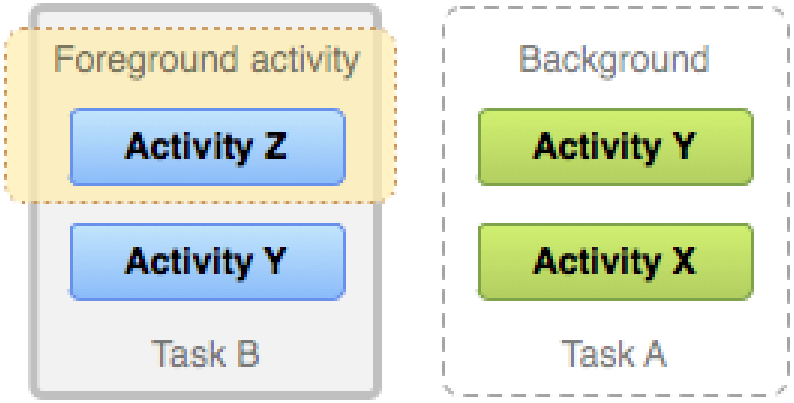
ACTIVITY LIFECYCLE AND TASK

- When the current activity starts another, the new activity is pushed on the top of the stack and takes focus.
- The previous activity remains in the stack, but is stopped



ACTIVITY AND TASK

- All activities belong to a task
- A task contains a collection of activities in the order in which the user interacts with them
- Tasks can move to the background and retain the state of each activity in order for users to perform other tasks without losing their work



INTENT

- Intents are asynchronous messages which allow Android components to request functionality from other components of the Android system.
- Intents can be used to signal to the Android system that a certain event has occurred.
- Other components in Android can register to this event via an intent filter.

INTENT

- Intent are sent to Android system by method call.
- For example:
  - An ***activity*** can send an ***intent*** to the Android system which starts another ***activity*** via the following code.

```
# Start the activity connect to the specified class

Intent i = new Intent(this, ActivityTwo.class);
i.putExtra("Value1", "This value one for ActivityTwo ");
i.putExtra("Value2", "This value two ActivityTwo");
startActivity(i);
```

- ActivityTwo will be called.

INTENT

- Can be used to call activities
  - startActivity(Intent)
- Starting services
  - startService(Intent)
- Retrieving result data from the called activity
  - startActivityForResult()
  - onActivityResult()
- Send broadcast
  - sendBroadcast()

SERVICE

- A facility for the application to tell the system about something it wants to be doing in the background (even when the user is not directly interacting with the application).
  - Context.startService(), which ask the system to schedule work for the service, to be run until the service or someone else explicitly stop it.
  - Service.onStartCommand(Intent intent, int flags, int startId)
- A facility for an application to expose some of its functionality to other applications.
  - Context.bindService(), which allows a long-standing connection to be made to the service in order to interact with it.
  - Service. onBind(Intent intent)

SERVICE

- A Service is **not** a separate process.
  - The Service object itself does not imply it is running in its own process; unless otherwise specified, it runs in the same process as the application it is part of.
- A Service is **not** a thread.
  - It is not a means itself to do work off of the main thread (to avoid Application Not Responding errors).

CONTENT PROVIDER

- Content providers are one of the primary building blocks of Android applications, providing content to applications.
- They encapsulate data and provide it to applications through the single ContentResolver interface.
- A content provider is only required if you need to share data between multiple applications. For example, the contacts data is used by multiple applications and must be stored in a content provider.

CONTENT PROVIDER

Provider	Since	Usage
Browser	SDK 1	Manages your web-searches, bookmarks and browsing-history.
CalendarContract	SDK 14	Manages the calendars on the user's device.
CallLog	SDK 1	Keeps track of your call history.
ContactsContract	SDK 5	Deals with all aspects of contact management.
MediaStore	SDK 1	The content provider responsible for all your media files like music, video and pictures.
Settings	SDK 1	Manages all global settings of your device.

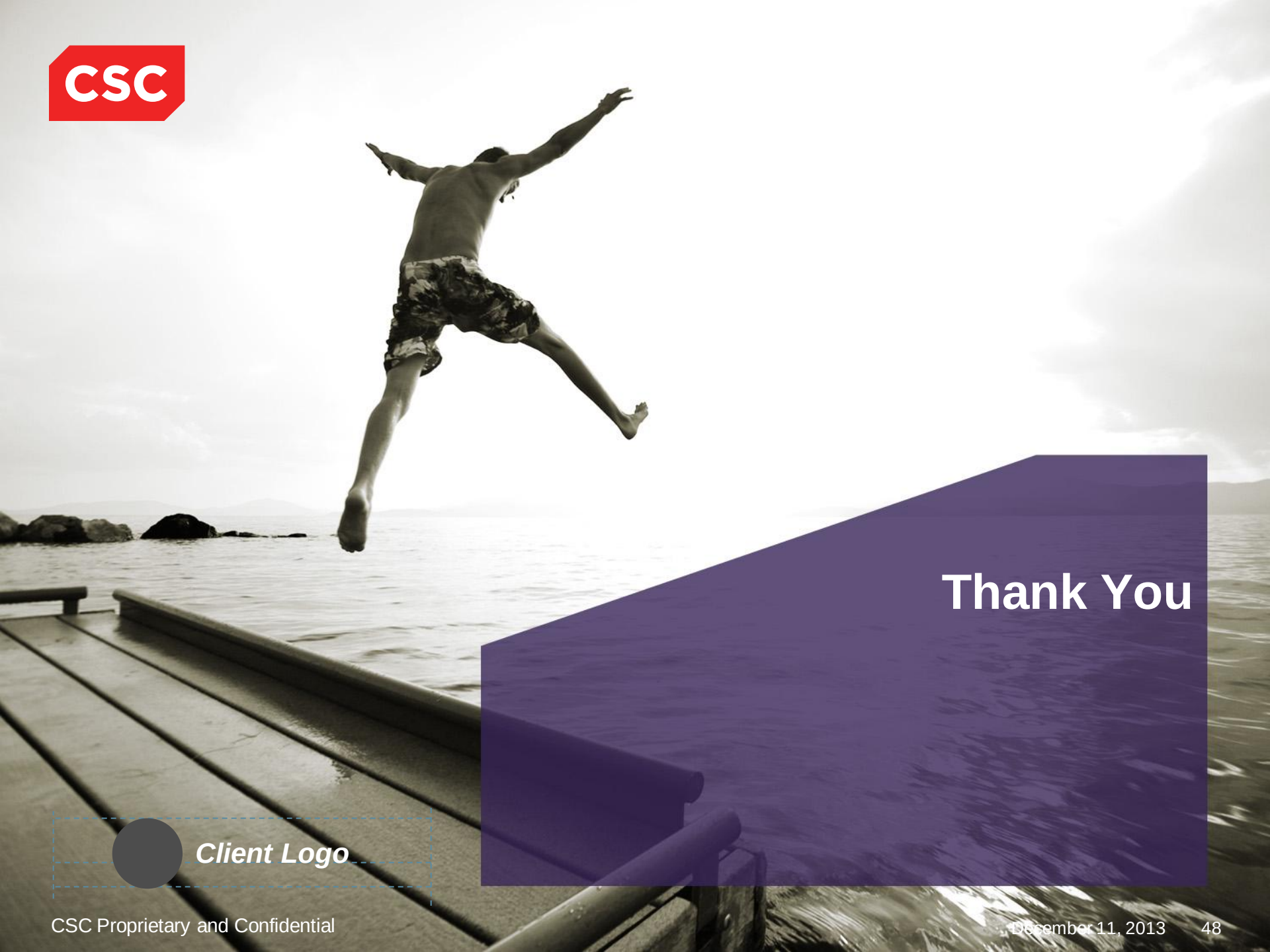
BROADCAST RECEIVER

- Base class for code that will receive intents sent by sendBroadcast().
- There are two major classes of broadcasts that can be received:

  – Normal broadcasts (sent with Context.sendBroadcast) are completely asynchronous. All receivers of the broadcast are run in an undefined order, often at the same time.

  – Ordered broadcasts (sent with Context.sendOrderedBroadcast) are delivered to one receiver at a time. As each receiver executes in turn, it can propagate a result to the next receiver, or it can completely abort the broadcast so that it won't be passed to other receivers. The order receivers run in can be controlled with the android:priority attribute of the matching intent-filter; receivers with the same priority will be run in an arbitrary order.

Q&A



Thank You

Client Logo

Revision History

Date	Version	Description	Updated by	Reviewed and Approved By
12 May 2013	0.5	Release for review	Nam Tu	


CSC
BUSINESS SOLUTIONS
TECHNOLOGY
OUTSOURCING