Short Introduction to Android

This document : https://arnaud.nauwynck.github.io/

Android..

android

Developer	Google Open Handset Alliance
Written in	Java (UI), C (core), C++ and more[1]
OS family	Unix-like
Working state	Current
Source model	Open source (most devices include proprietary components, such as Google Play)
Initial release	September 23, 2008; 9 years ago[2]
Latest release	8.1.0 "Oreo" / December 5, 2017; 3 months ago
Latest preview	Android P / March 7, 2018; 20 days ago[3]
Marketing target	Smartphones, tablet computers, smartTVs (Android TV), Android Auto and smartwatches (Wear OS)
Available in	100+ languages[4]
Package manager	APK (primarily through Google Play; installation of APKs also possible locally or from alternative sources such as F-Droid)
Platforms	32- and 64-bit ARM, x86, x86-64, MIPS and MIPS64
Kernel type	Monolithic (modified Linux kernel)
Userland	Bionic libc,[5] mksh shell,[6] Toybox as core utilities beginning with Android 6.0,[7][8] previously native core utilities with a few from NetBSD[9][10]
Default user interface	Graphical (multi-touch)
License	Apache License 2.0 GNU GPL v2 for the Linux kernel modifications[11]
Official website	android.com

Article	Talk		Read	View sou...

Android (operating system)

From Wikipedia, the free encyclopedia

For other things called "android", see Android (disambiguation).

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

https://developer.android.com/

Android Developers
https://developer.android.com/index.html ▼
The official site for **Android developers**. Provides the **Android** SDK and documentation for app **developers** and designers.

Develop Apps
Guides - Download Android Studio - Android Design - Distribute - ...

Download Android Studio
Download the official Android IDE and developer tools to build ...

Developer Guides
Welcome to the Android developer guides. The documents listed in ...

Google Play Console
Success
quality.

Android Design
Android uses a new design metaphor inspired by paper and ...

Wear
The off
Develo

More results from android.com »

Open navigation   ...velopers   DESIGN   **DEVELOP**   DISTRIBUTE

← Guides

Develop > Guides > Build Your First App

Create an Android Project

This lesson shows you how to create a new Android project with Android Studio and describ

First, be sure you have installed the latest version of Android Studio. Download Android Studio here.

1. In the **Welcome to Android Studio** window, click **Start a new Android Studio project**.

App Basics

Introduction

Build Your First App   ⌃

Create an Android Project

Run Your App

Build a Simple User Interface

Start Another Activity

App Fundamentals

App Resources   ⌄

App Manifest File   ⌄

App Permissions   ⌄

Devices

Device Compatibility   ⌄

Wear   ⌄

Welcome to Android Studio

Android Studio
Version 3.0

☀ Start a new Android Studio project
🗁 Open an existing Android Studio project
⬇ Check out project from Version Control ▾
⊞ Profile or debug APK
⬏ Import project (Gradle, Eclipse ADT, etc.)
⬐ Import an Android code sample

⚙ Configure ▾   Get Help ▾

Or if you have a project opened, select **File > New Project**.

https://source.android.com

Training

https://developers.google.com/training/android/

Source   Setup   Security   Porting   Tuning   Compatibility   ❯   🔍 Search   GO TO CODE
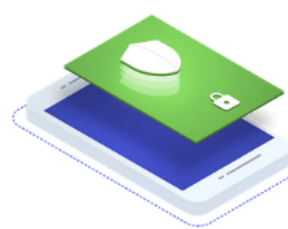
Android unites the world! Use the open source Android operating system to power your device.

GET SOURCE
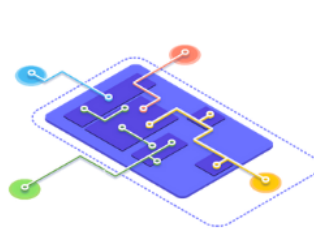
8.1 interfaces and architecture

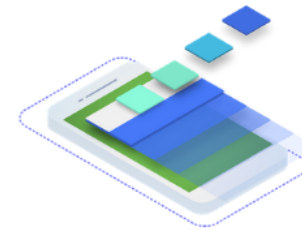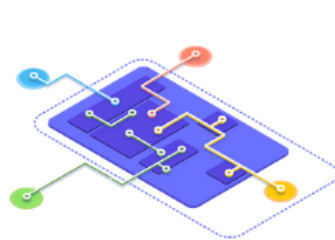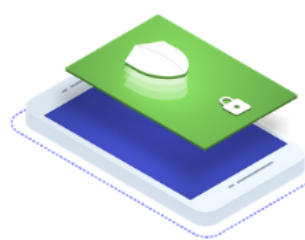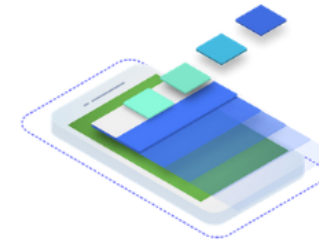Port the latest Android platform using simple HIDL interfaces to create compelling devices for your customers.

LEARN TREBLE

Securing Android is essential

Find out how the Android security program works and learn how to implement the latest features.

IMPLEMENT SECURITY

Get compatible, get apps

Offer a consistent experience with other Android-powered devices and get the ability to include more apps.

TEST DEVICES

About the Android Open Source Project

Android is an open source software stack for a wide range of mobile devices and a corresponding open source project led by Google. This site and the Android Open Source Project (AOSP) repository offer the information and source code you need to create custom variants of the Android stack, port devices and accessories to the Android platform, and ensure your devices meet compatibility requirements.

https://www.tutorialspoint.com/android

History

2003
Startup..

2005
bought by
Google

2010
Nexus
phone

2014
Android TV

Wear OS

androidauto

Architecture Overview

Core Concept 1 :

Linux
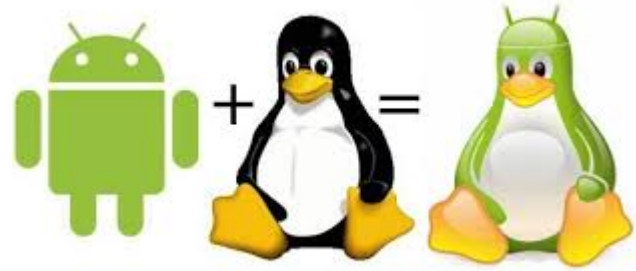
Base Linux Kernel Security

System User Account (uid/gid)

All app files are protected by unix chmod
All process run with current uid

Security Permission checked in System Calls
= standard kernel linux security

"Root" user 0 can do everything …
Only Android low-level services run as root

1 Running App = 1 Process
(Linux JVM)

Start App → **Running JVM** → Gracefull App Exit
or
Kill App

Adb Shell .. PS

Sample User App (Calculator)

Calculator

```
shell       8645  1      4700     200    poll_sched  00000000 R /sbin/adbd
u0_a79      8651  246    966444   15252  SyS_epoll_  00000000 S com.android.gallery3d
u0_a19      8740  246    655600   26272  SyS_epoll_  00000000 S com.android.contacts
u0_a10      8814  246    656132   30400  SyS_epoll_  00000000 S com.ape.calculator2
shell       8845  8645   1984     828    sigsuspend  b6e630cc S /system/bin/sh
root        8870  2      0        0      worker_thr  00000000 S kworker/u8:2
root        8871  2      0        0      worker_thr  00000000 S kworker/u8:4
shell       8892  8845   2156     712            0   b6dce920 R ps
```

UID = u0_a10

```
shell@V2510:/system/app $ ls
ApeCalculator
ApeMyosVersion
ApeResetUserData
ApeSaletracker
ApeSystemUpdate
ApeTorch
ApnSettingsPlugin
```

```
shell@V2510:/system/TLUI $ ls
com.ape.calculator2
com.ape.filemanager
com.ape.launcher3
com.ape.led
com.ape.music
com.ape.myseneschal
```

```
shell@V2510:/system/app/ApeCalculator $ find
.
./ApeCalculator.apk
./oat
./oat/arm
./oat/arm/ApeCalculator.odex
```
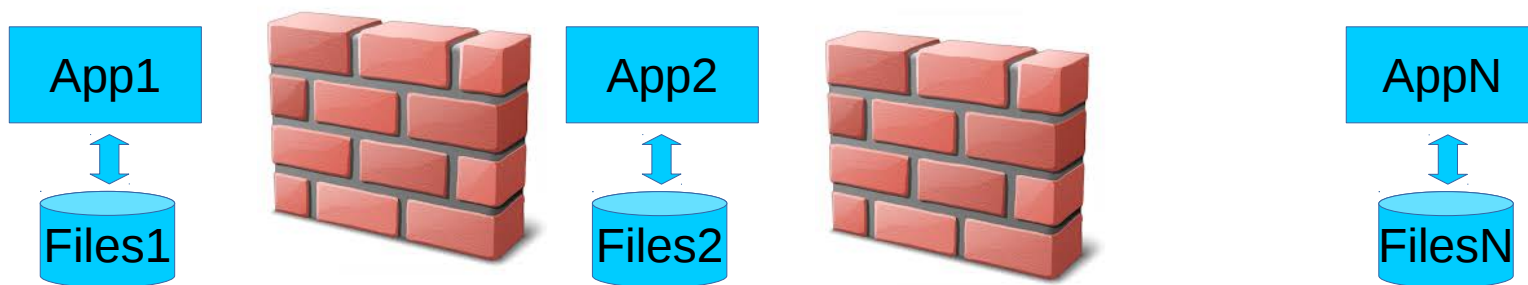
1 Installed App = 1 UID

Install 1 App => create a new UID !!

Files owned by 1 App => files owned by UID

All Apps (Files) are ISOLATED

App1 ↕ Files1

App2 ↕ Files2

AppN ↕ FilesN

No "Root" access

Apps are installed by "apk" / MarketStore
 … similar to "sudo apt-get install "

BUT
- forcing a new uid/gid for each app
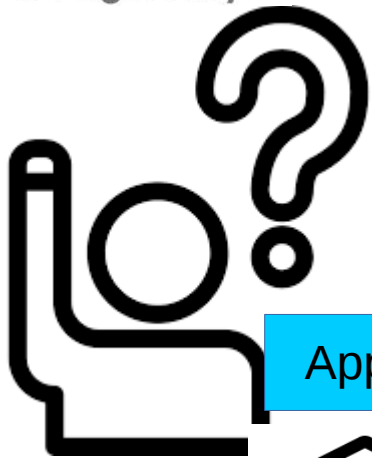- apps (files) are not owned to "root"
- no command "sudo", no root acess

 No apps can be root, only by jailbreaking your phone

Installing App = Accepting "Manifest" Permissions

Each App as a "Manifest" Permission required files
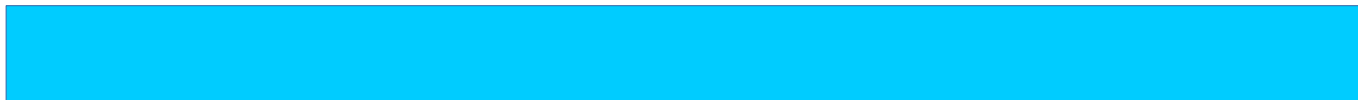When You install = You accept permissions

App1

App2

AppN

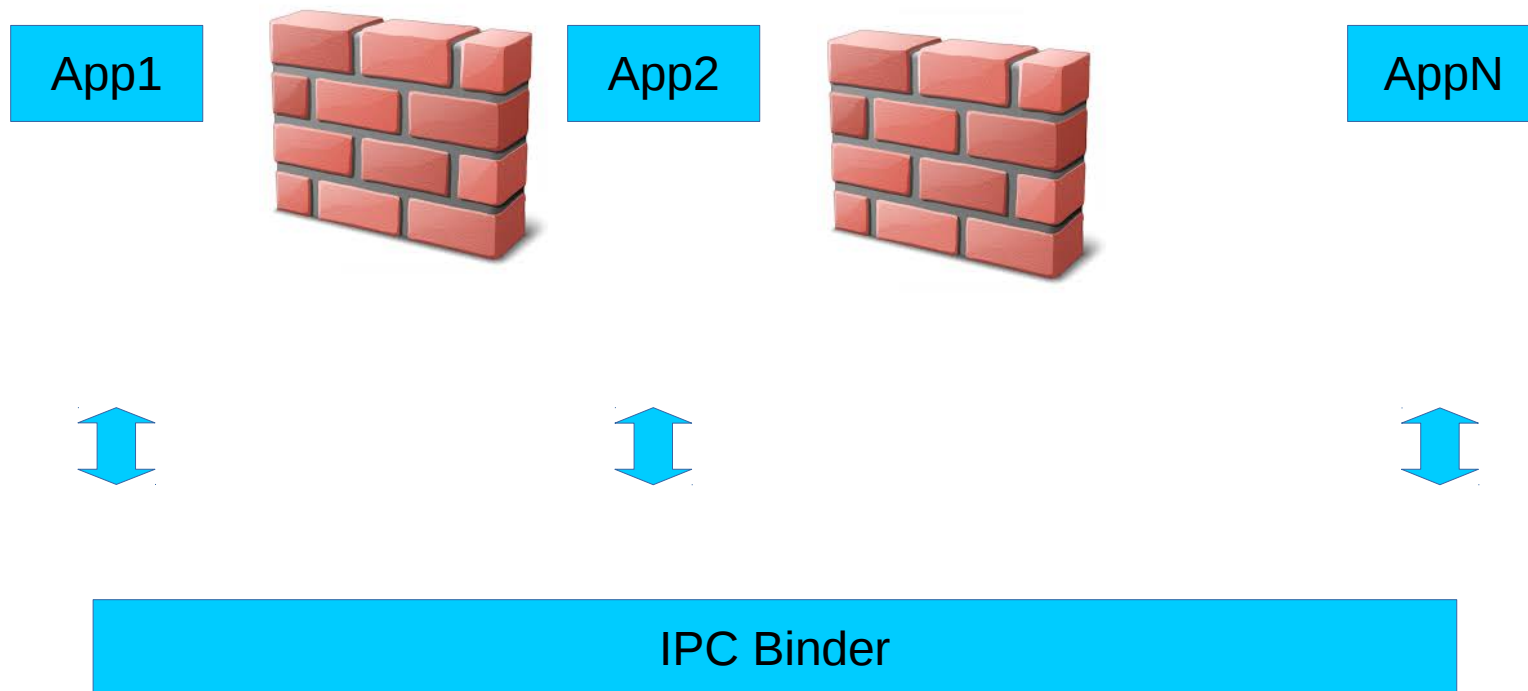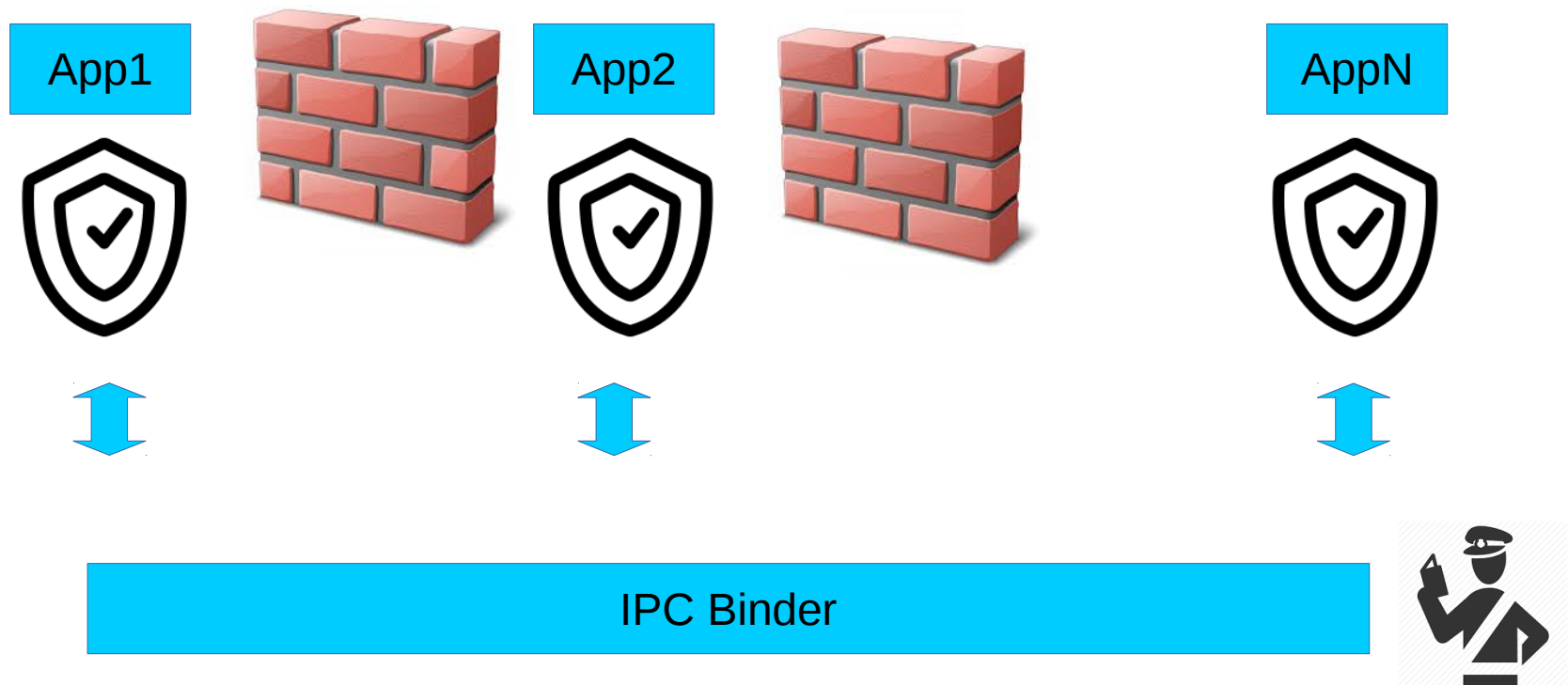# Core Concept 2

Communications between Apps

Communication between Apps
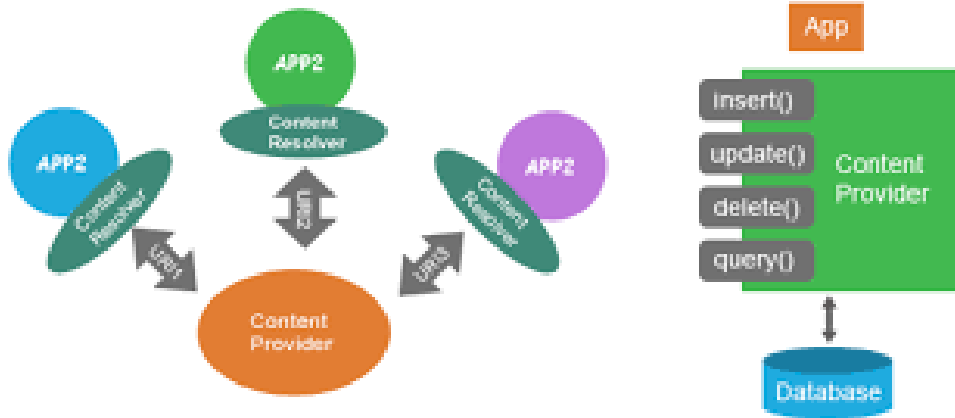= Using IPC Binder messages (service)

App1

App2

AppN

IPC Binder

Permissions + Communications between Apps

IPC Binder messages (service)
+ Permissions (Manifest file per App)

Apps Isolations / Data
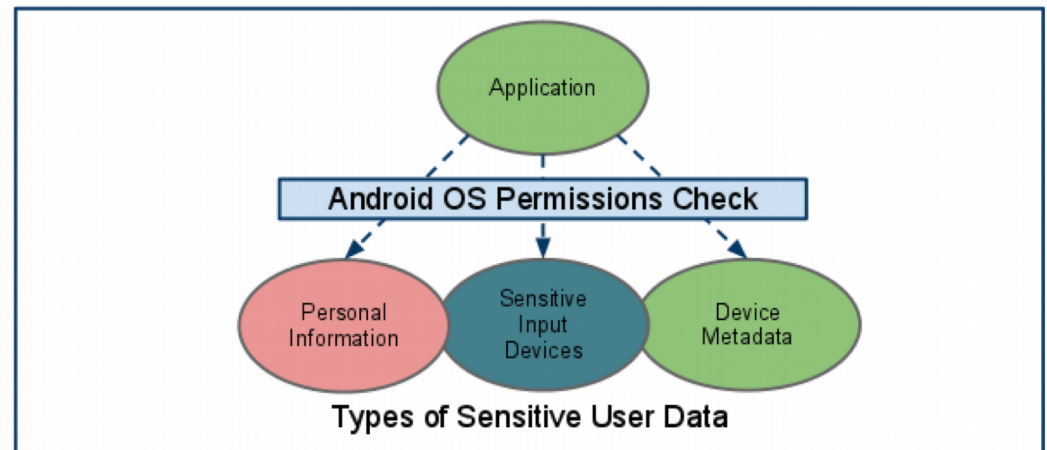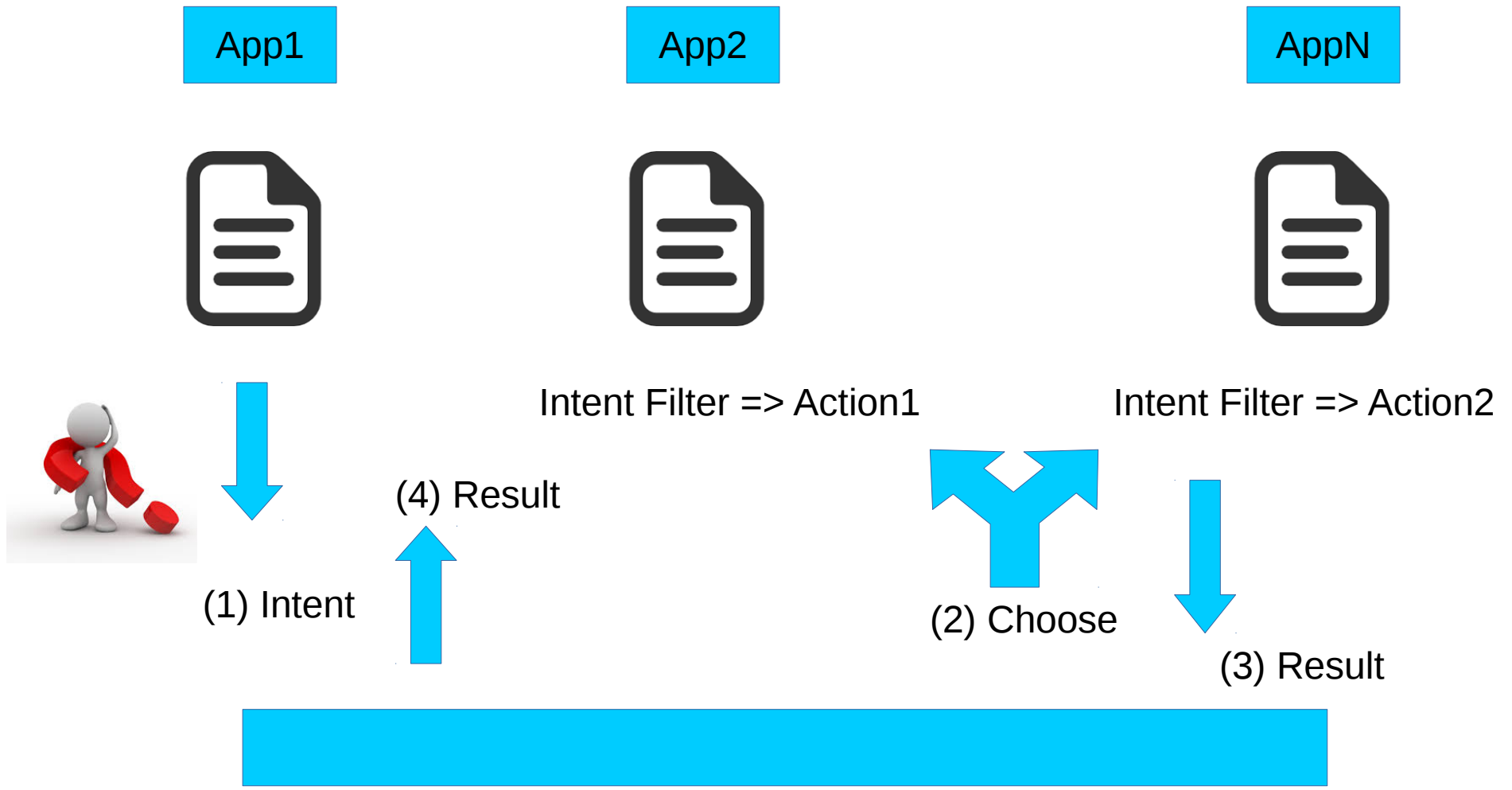

Data Between Apps

System Data

Core Concept 3 : Intent

INTENT   ACTIONS   RESULTS

Your INTENT influences your ACTIONS which influences your RESULTS.

Manifest – Intent Capabilities

App1

App2

AppN

Intent Filter => Action1

Intent Filter => Action2

(4) Result

(1) Intent

(2) Choose

(3) Result

Manifest – Intent Filter

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

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="com.example.myapplication.MainActivity"
            android:label="@string/app_name" >

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>

    </application>
</manifest>
```
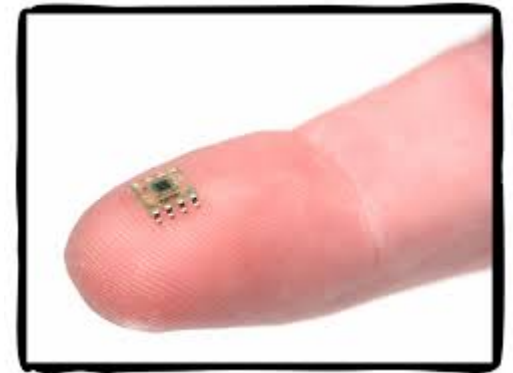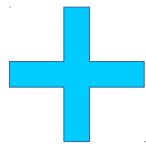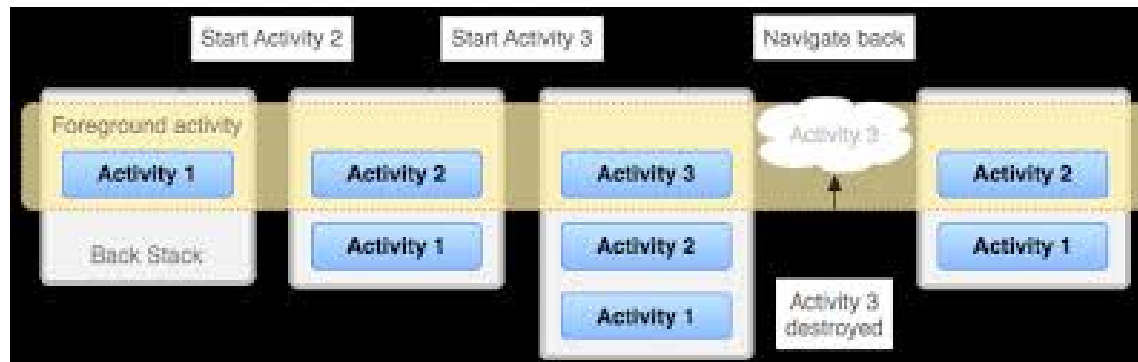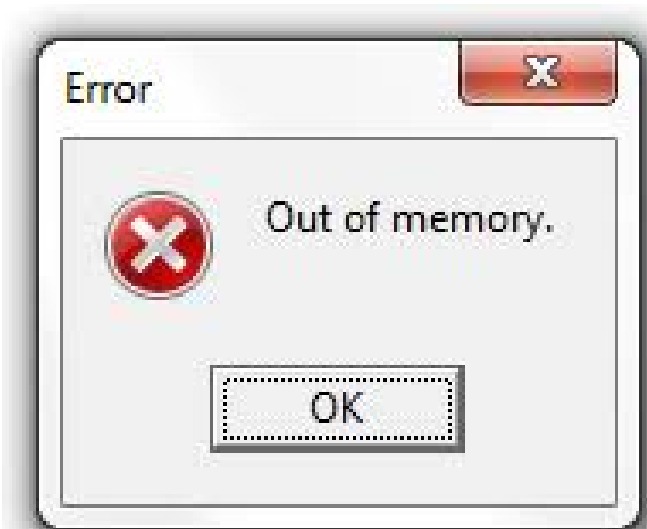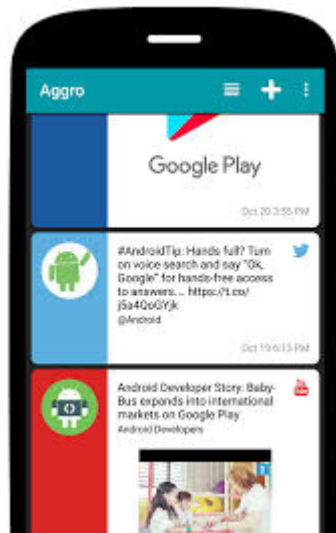
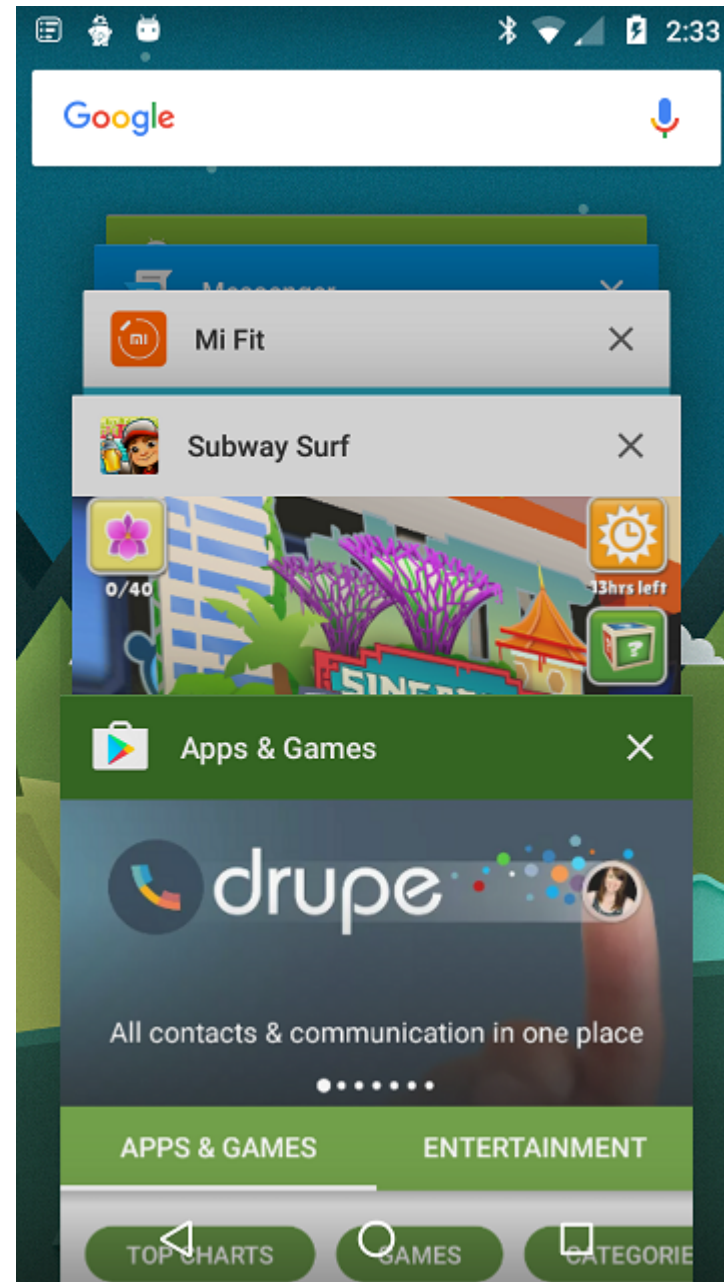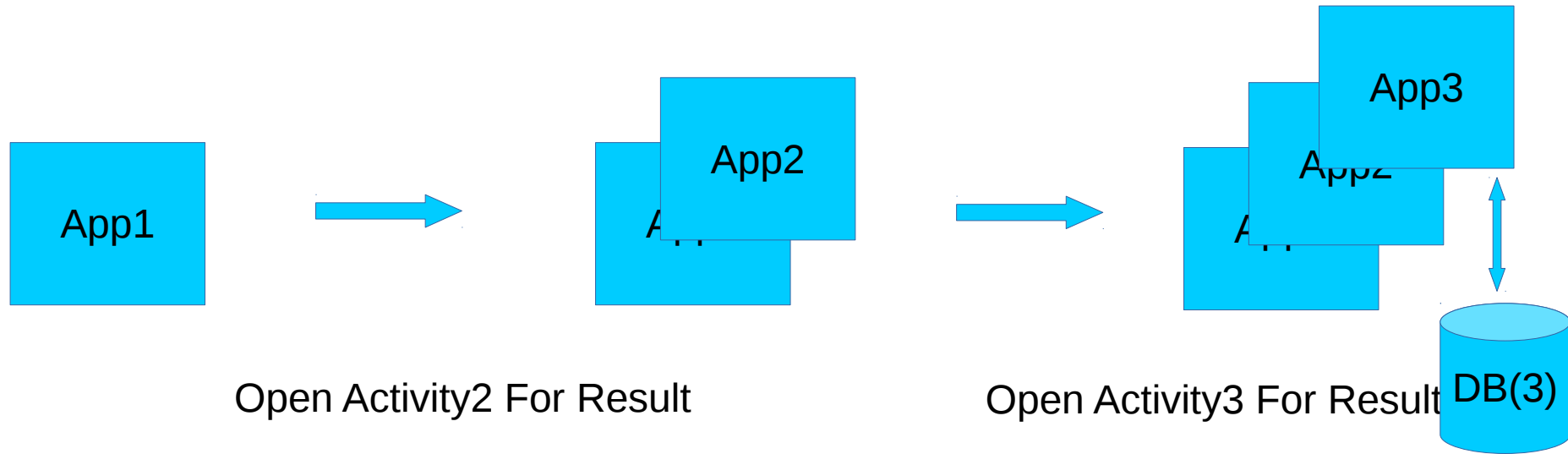# Core Concept 3 : RAM
Start / Kill / Resume App

Multiple Activities – Small RAM ...

Activities Stack & Recent

BACK   HOME   RECENTS
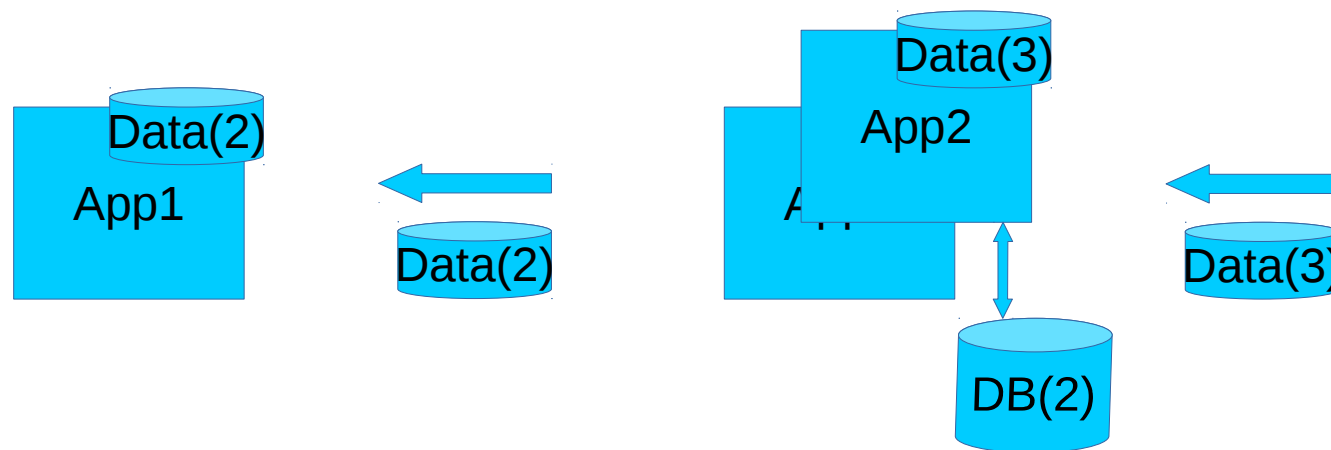
Activities Stack
(Open Modal Dialog.. Return)

App1 → App2 / App1

Open Activity2 For Result

example:
take photo
=> open "Photo" application

App3 / App2 / App1 ⟷ DB(3)

Open Activity3 For Result

example:
send.. select Contact
  => open "Contact" application

Data(2) App1 ← Data(2) ← Data(3) App2 / App1 ⟷ DB(2) ← Data(3)

Activity Memento

Start App

Run
… live data

App1

Save Memento

App1

Kill App

(re)Start App

App1

Resume...

App1

Activity State Diagram
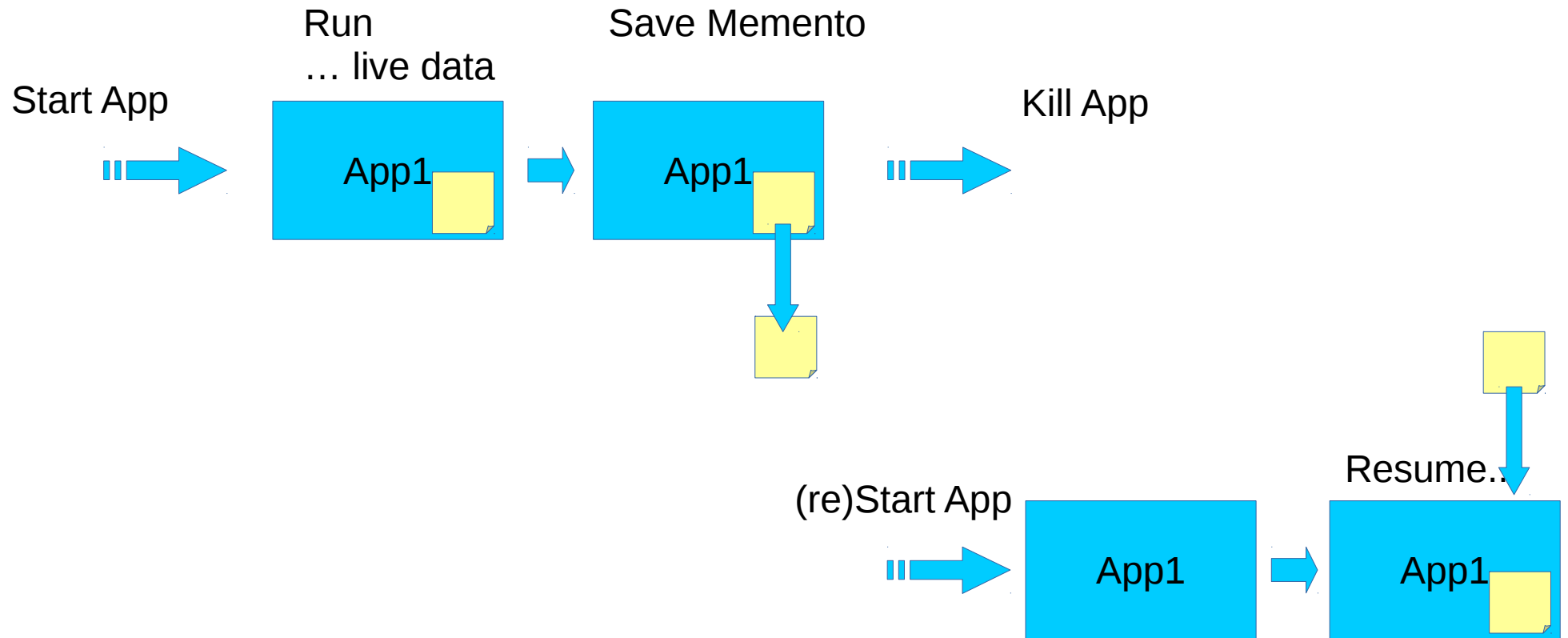
Visible Activity State Diagram ...

Fast JVM Start Time

Classical

Running JVM

Android Dalvik

Zigote JVM (suspended)

Fork Linux Process (FAST)

Zigote JVM (suspended)

Resume Process

Load Class + call main()

Running JVM

Iphone Cheating

Display Screenshot

Running Process
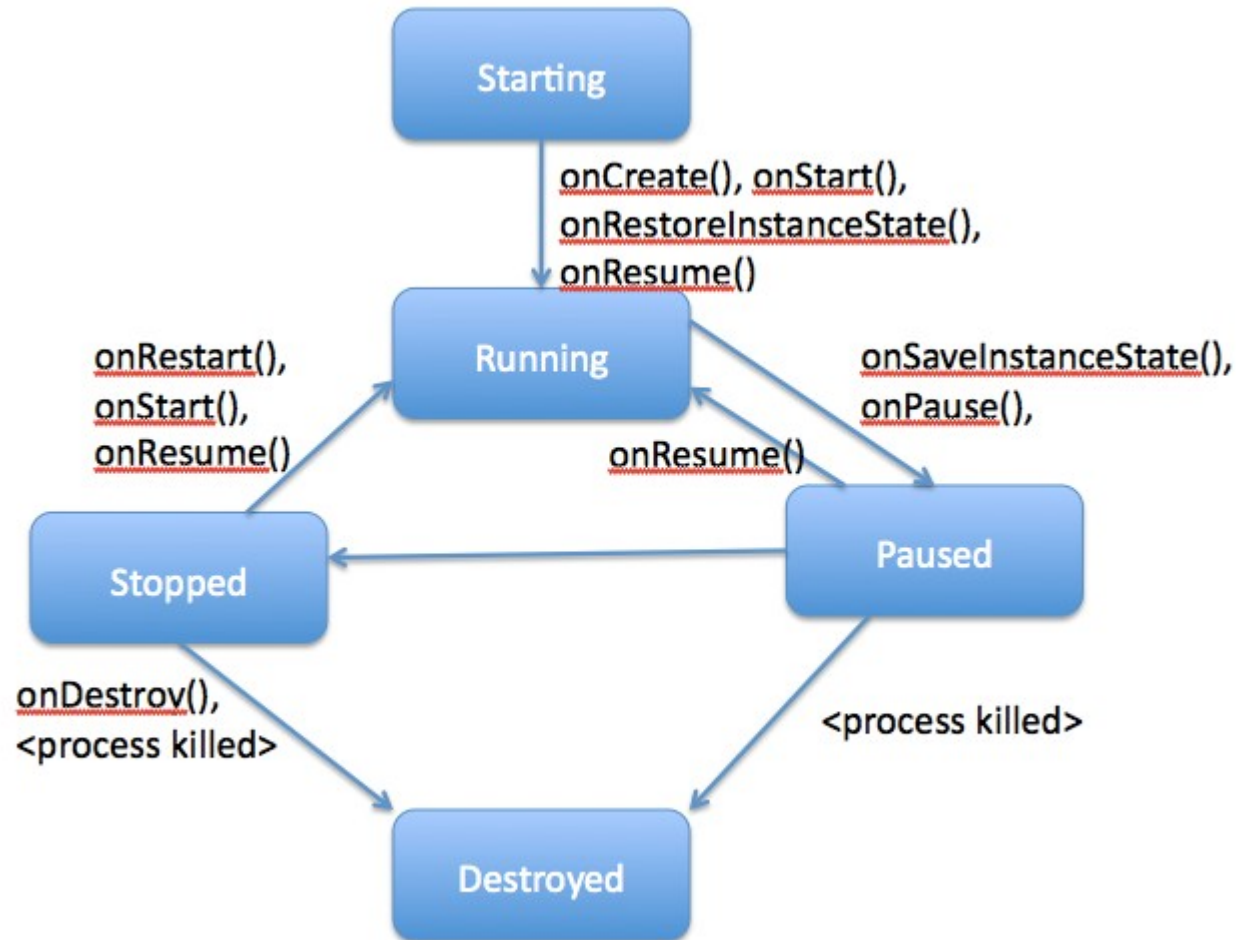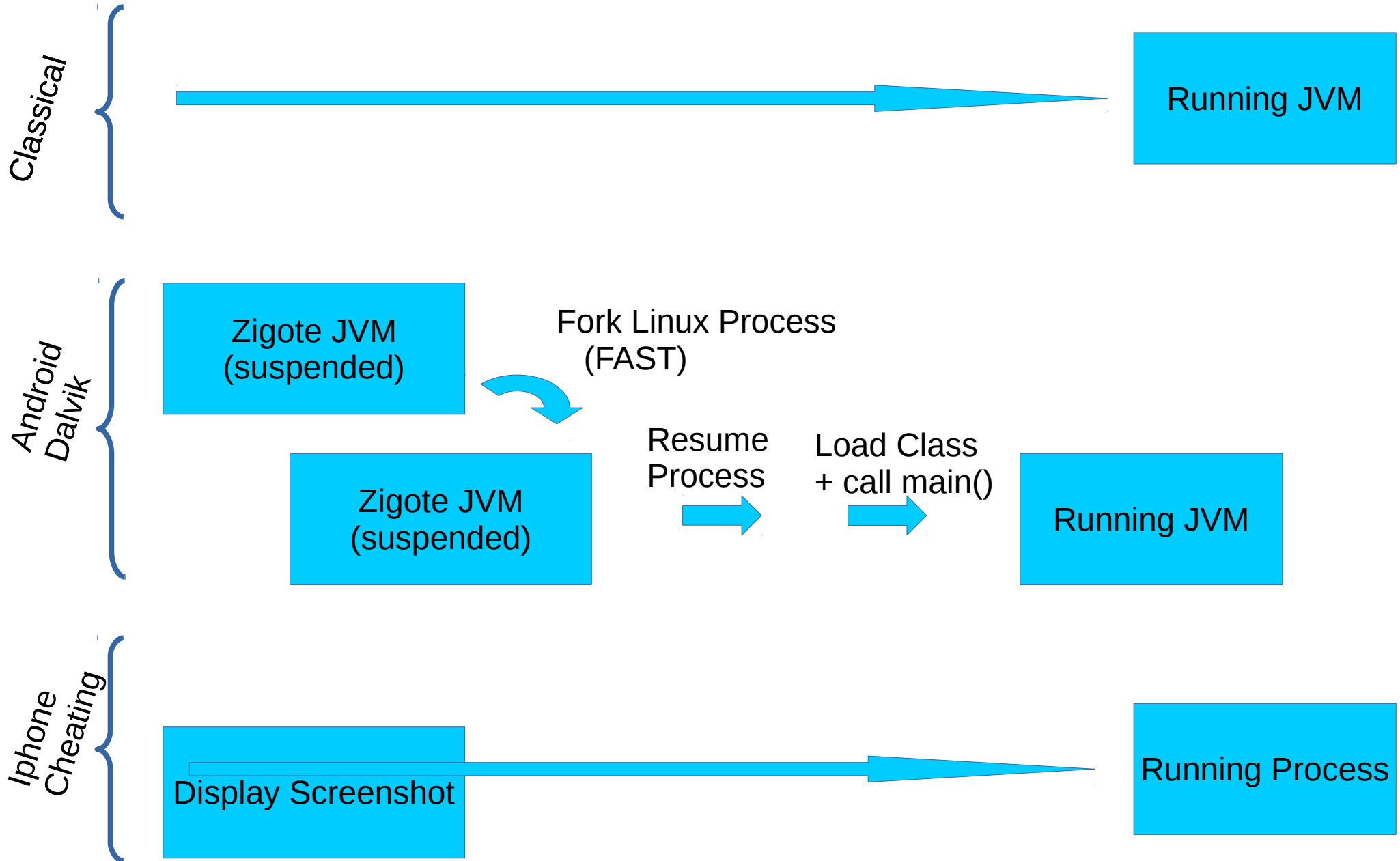
Core Concept : Framework / APIs

HAL = Hardware Abstraction Layer

https://source.android.com/devices/architecture/hal
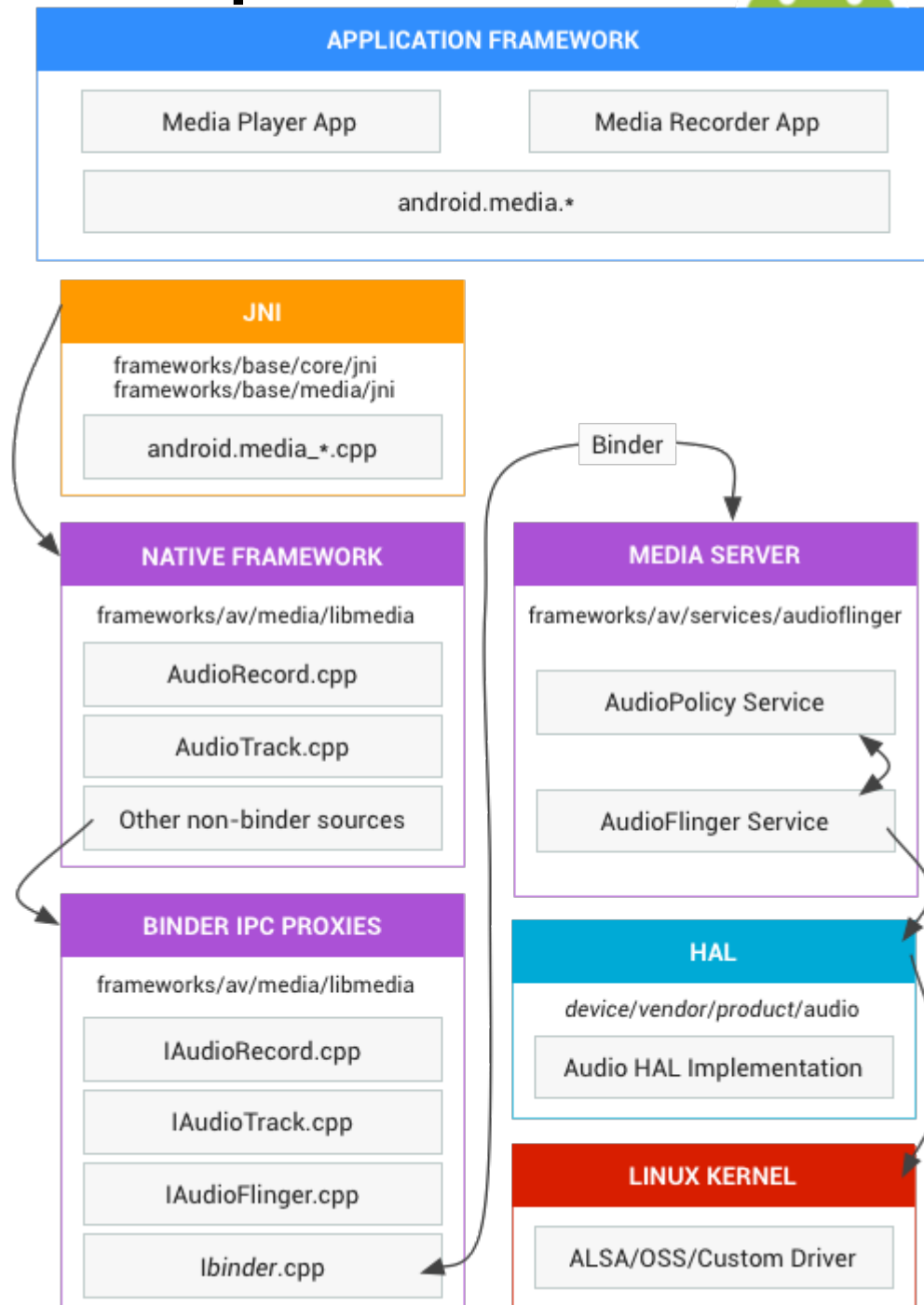
Hardware Abstraction Layer (HAL)

A HAL defines a standard interface for hardware vendors to implement, which enables Android to be agnostic about lower-level driver implementations. Using a HAL allows you to implement functionality without affecting or modifying the higher level system. HAL implementations are packaged into modules and loaded by the Android system at the appropriate time.
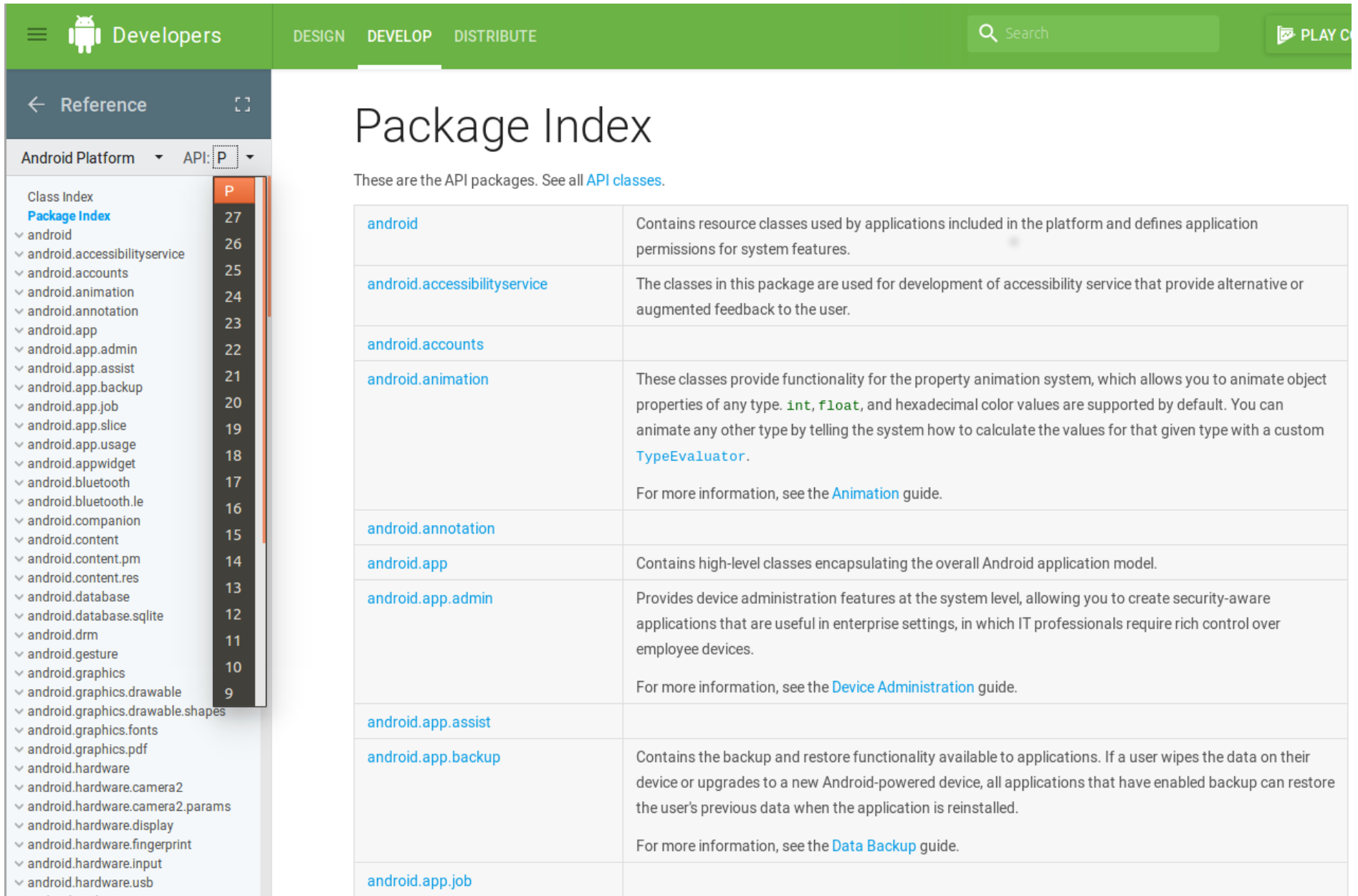
Hardware Abstraction Layer (HAL)

AUDIO	AUTOMOTIVE	BLUETOOTH	CAMERA	DRM	GRAPHICS
INPUT	MEDIA	PERIPHERALS	SENSORS	STORAGE	TV

Kernel < HAL < API
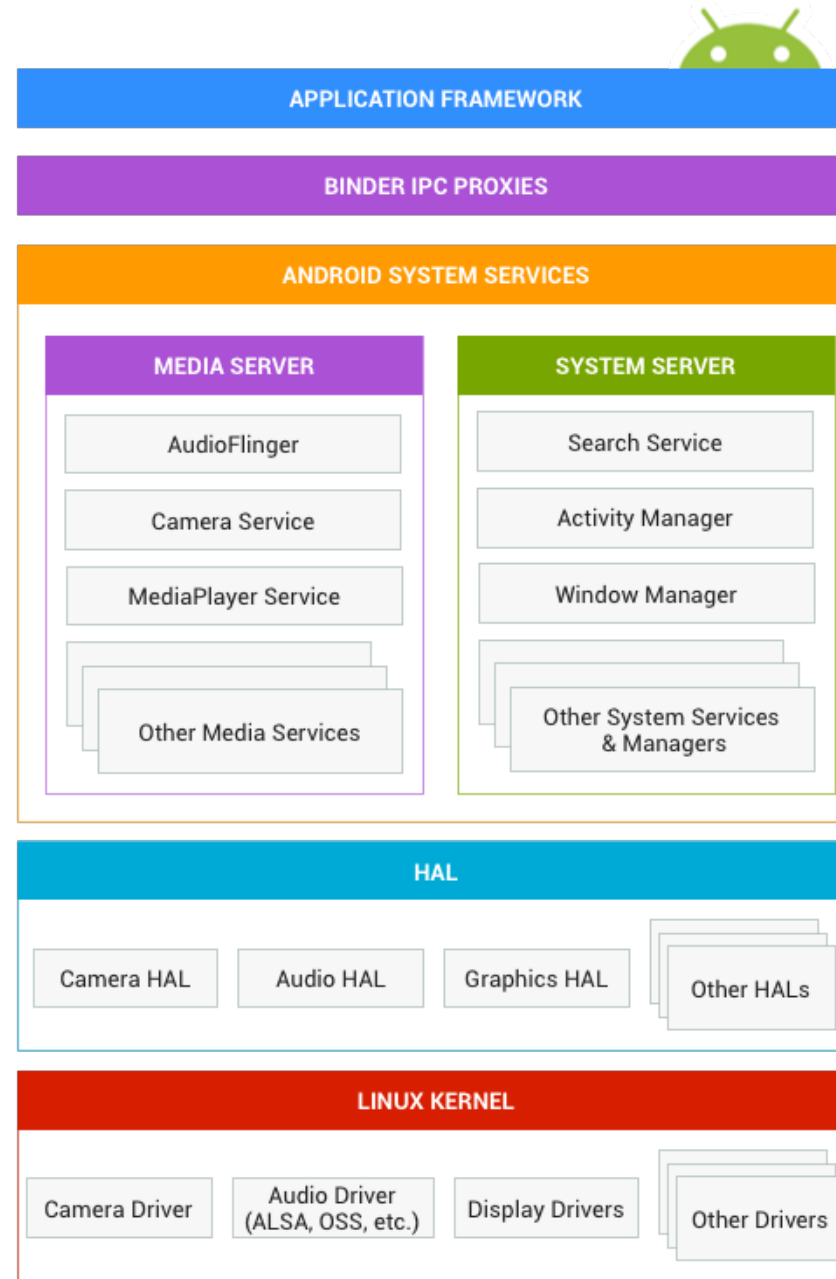
Example: Audio

Core Concept
API – Services

APIs … (current API level=27)

https://developer.android.com/reference/

Example API : Text To Speech

https://developer.android.com/reference/android/speech/tts/TextToSpeech.html

Example TTS Code ..

```java
TextToSpeech tts =
    new TextToSpeech(getApplicationContext(),
        new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if(status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.UK);
                }
            }
        });


String toSpeak = "Hello Android";

tts.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null);
```

Core Concept
Services / Job API – No Threads

Core Concept : UI

Resources
+ Layouts
+ Components
+ Fragments
+ ...

Defining UI Resources

Simple Resources

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="red">#f00</color>
    <color name="highlight">@color/red</color>
</resources>
```

```
MyProject/
    src/
        MyActivity.java
    res/
        drawable/
            graphic.png
        layout/
            main.xml
            info.xml
        mipmap/
            icon.png
        values/
            strings.xml
```

Layout Resources

In Java: `R.layout.filename`

In XML: `@[package:]layout/filename`

SYNTAX:

```xml
<?xml version="1.0" encoding="utf-8"?>
<ViewGroup
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@[+][package:]id/resource_name"
    android:layout_height=["dimension" | "match_parent" | "wrap_content"]
    android:layout_width=["dimension" | "match_parent" | "wrap_content"]
    [ViewGroup-specific attributes] >
    <View
        android:id="@[+][package:]id/resource_name"
        android:layout_height=["dimension" | "match_parent" | "wrap_content"]
        android:layout_width=["dimension" | "match_parent" | "wrap_content"]
        [View-specific attributes] >
        <requestFocus/>
    </View>
    <ViewGroup >
        <View />
    </ViewGroup>
    <include layout="@layout/layout_resource"/>
</ViewGroup>
```

Using Resources in Java

[<package_name>.]R.<resource_type>.<resource_name>

```java
// Load a background for the current screen from a drawable resource
getWindow().setBackgroundDrawableResource(R.drawable.my_background_image) ;

// Set the Activity title by getting a string from the Resources object, because
//   this method requires a CharSequence rather than a resource ID
getWindow().setTitle(getResources().getText(R.string.main_title));

// Load a custom layout for the current screen
setContentView(R.layout.main_screen);

// Set a slide in animation by getting an Animation from the Resources object
mFlipper.setInAnimation(AnimationUtils.loadAnimation(this,
        R.anim.hyperspace_in));

// Set the text on a TextView object using a resource ID
TextView msgTextView = (TextView) findViewById(R.id.msg);
msgTextView.setText(R.string.hello_message);
```

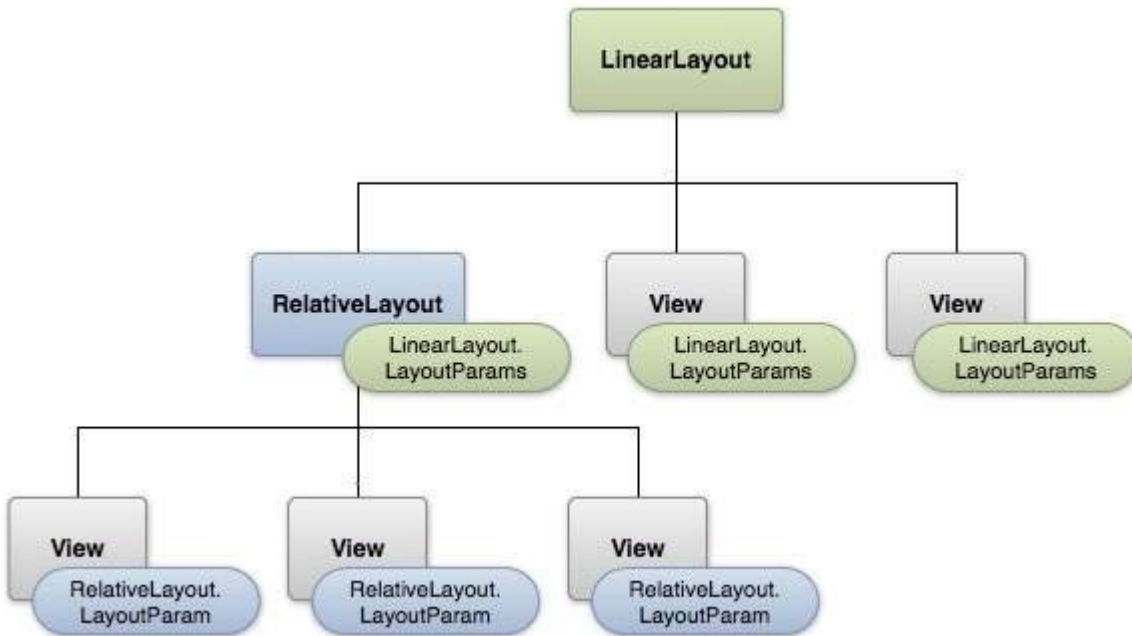# Using Resources in Xml

```
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/submit" />
```

Syntax

Here is the syntax to reference a resource in an XML resource:

@[<package_name>:]<resource_type>/<resource_name>

UI Layouts


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is a TextView" />

    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is a Button" />

    <!-- More GUI components go here  -->

</LinearLayout>
```
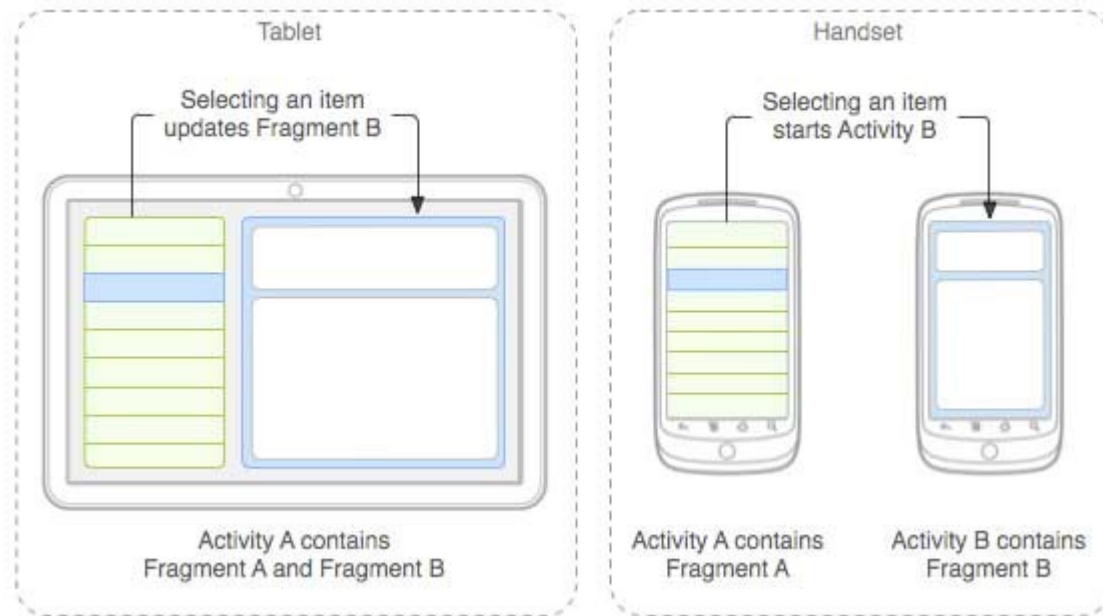
UI Fragments

Questions