CS 473 – MDP
Mobile Device Programming

Maharishi International University

CS 473 – MDP
Mobile Device Programming

MS.CS Program

Department of Computer Science

Renuka Mohanraj, Ph.D.,

Maharishi International University

CS 473 – MDP
Mobile Device Programming

Lesson 1

Introduction to Android

Maharishi International University

Wholeness of this lesson

This lecture serves as an introduction to Android Programming. We will briefly cover the basics of Android such as Architecture, Features, Releases and the Components. This will give us a solid foundation before creating android applications. *Similarly, why we want to start the day with TM; to make sure our mind is clear and rested before starting activity.*

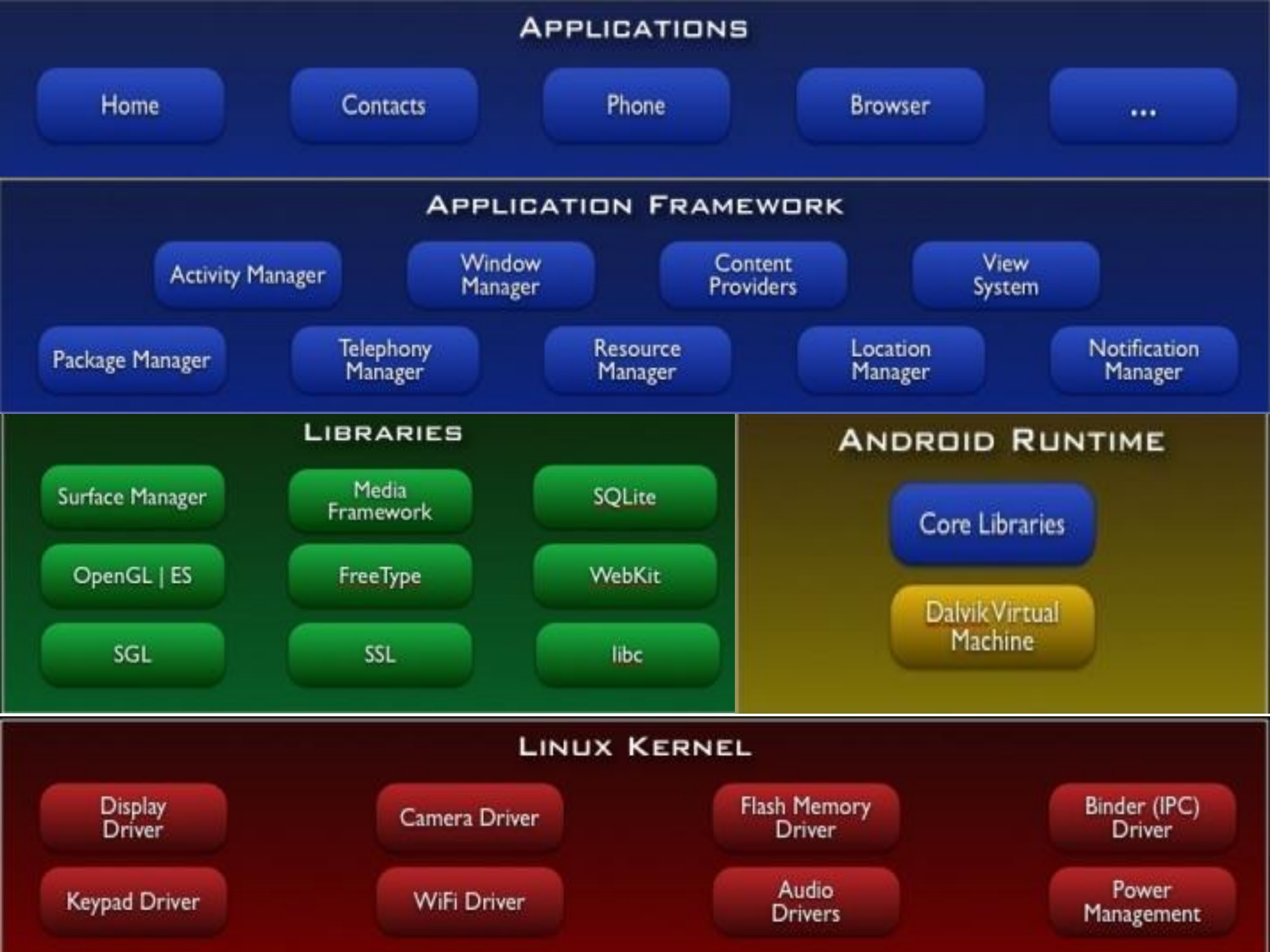
Agenda

- What is Android?
- Android Architecture
- Android Features
- Android History and Releases
- Android Components

What is Android?

- Android is a Mobile Platform which consist of OS(Linux), Middleware (Libraries and Application Framework) and Key Applications.
  - Mobile operating system based on Linux kernel
  - User Interface for touch screens
  - Used on over 80% of all smartphones
  - Powers devices such as Watches, TVs, and Cars[Android auto App]
  - Over 2 Million Android apps in Google Play store
  - Highly customizable for devices / by vendors
  - Open source
  - To know more about What is Android Click the Link: https://www.android.com/what-is-android/

Android Architecture



Applications

Home
Contacts
Phone
Browser
...

Application Framework

Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager

Libraries

Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc

Android Runtime

Core Libraries
Dalvik Virtual Machine

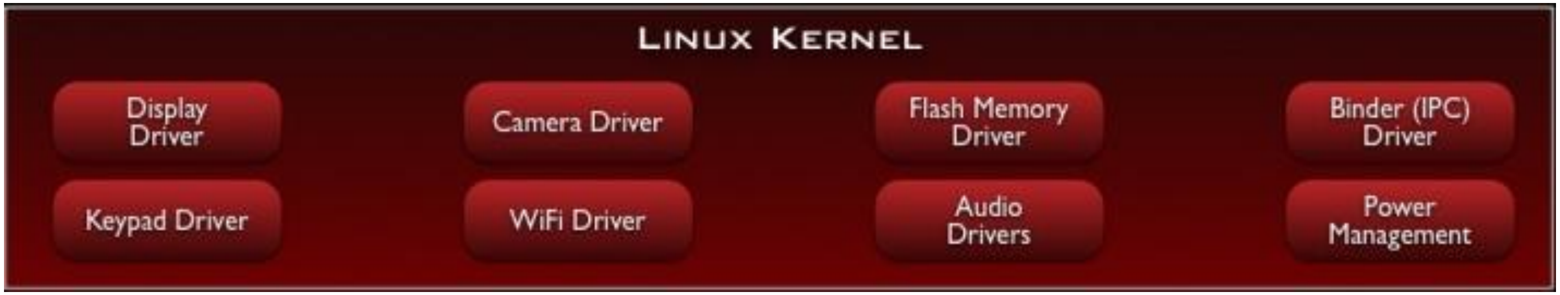
Linux Kernel

Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Android Architecture

- The Android architecture consists of the following four layers and divided into five sections:

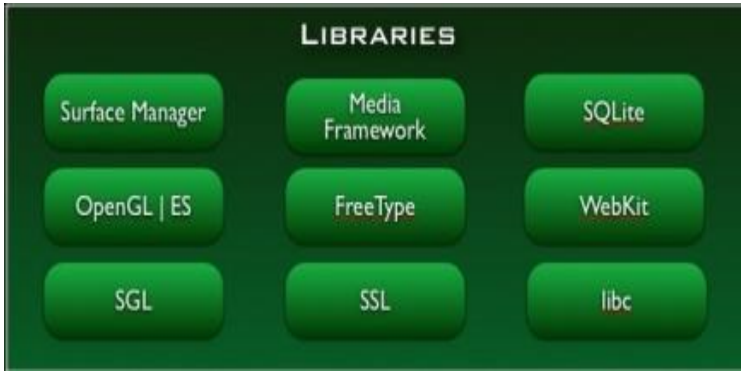
1. Linux Kernel – Hardware and Software interaction. It provides Security, Process management, Memory management, Device management and Multitasking.
It is also responsible for a level of abstraction between device hardware and upper layers of Android architecture. It consists of device drivers like camera, flash memory, Display, keypad, Wifi etc. IPC stands for Inter-Process Communication.

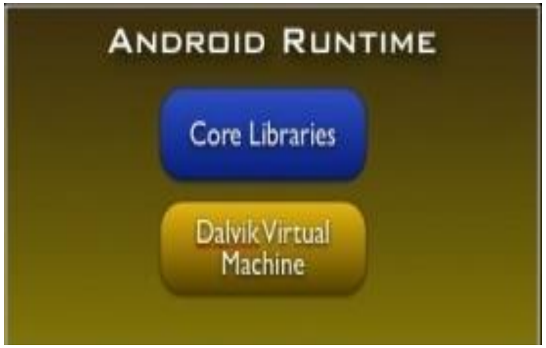


Android Architecture

2. Libraries and Android Runtime

- In addition to a set of standard Java development libraries, the Android development environment also includes the Android Libraries.
- (Native language(C/C++) libraries for other support – OpenGL – Graphics Library to display.
  - **SGL** stands for "Scalable Graphics Library“
  - **SSL** stands for “Secure Socket Layer”
  - ES – Embedded System
- Android Runtime(ART) : Meets the need of running in an embedded environment.





Android Architecture

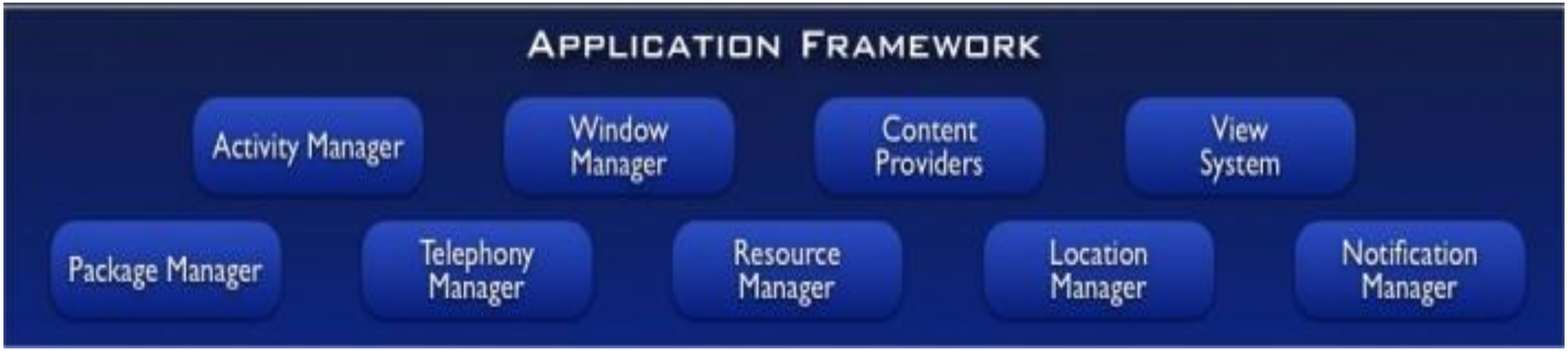
- DALVIK VIRTUAL MACHINE
  - Android uses DVM to optimize battery life, memory and performance.
  - It is like Java uses Java Virtual Machine (JVM).
  - The JVM runs .class files whereas the Dalvik VM runs .dex files, which are tailored to provide higher efficiency.
  - The Dalvik VM provides support for security, memory management, isolation, and threading.
  - DVM allows every Android application to run as a separate process, but with its own instance of DVM.
  - DVM is a virtual machine that is created when a process starts and is destroyed when the process exits

`

Android Architecture

3. **Application Framework** :

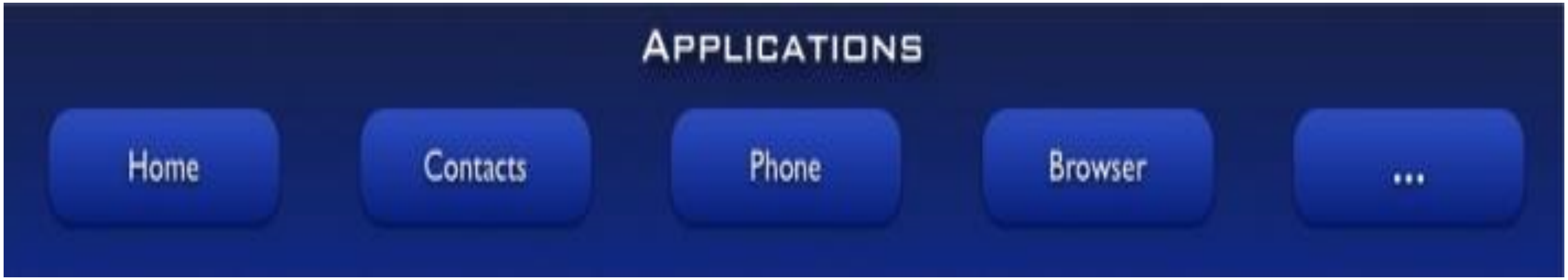
- Provides a readymade library for doing several task. (Eg : GPS, Wi-Fi,Bluetooth etc.,)



Android Architecture

4. **Applications :**

- At this top layer are the applications that ship with the Android device(such as Phone, Contacts, Browser, and so on), as well as applications that you download and install from the Android Market i.e. Google Play Store.
- Any applications that you develop are located at this layer.



Android Features

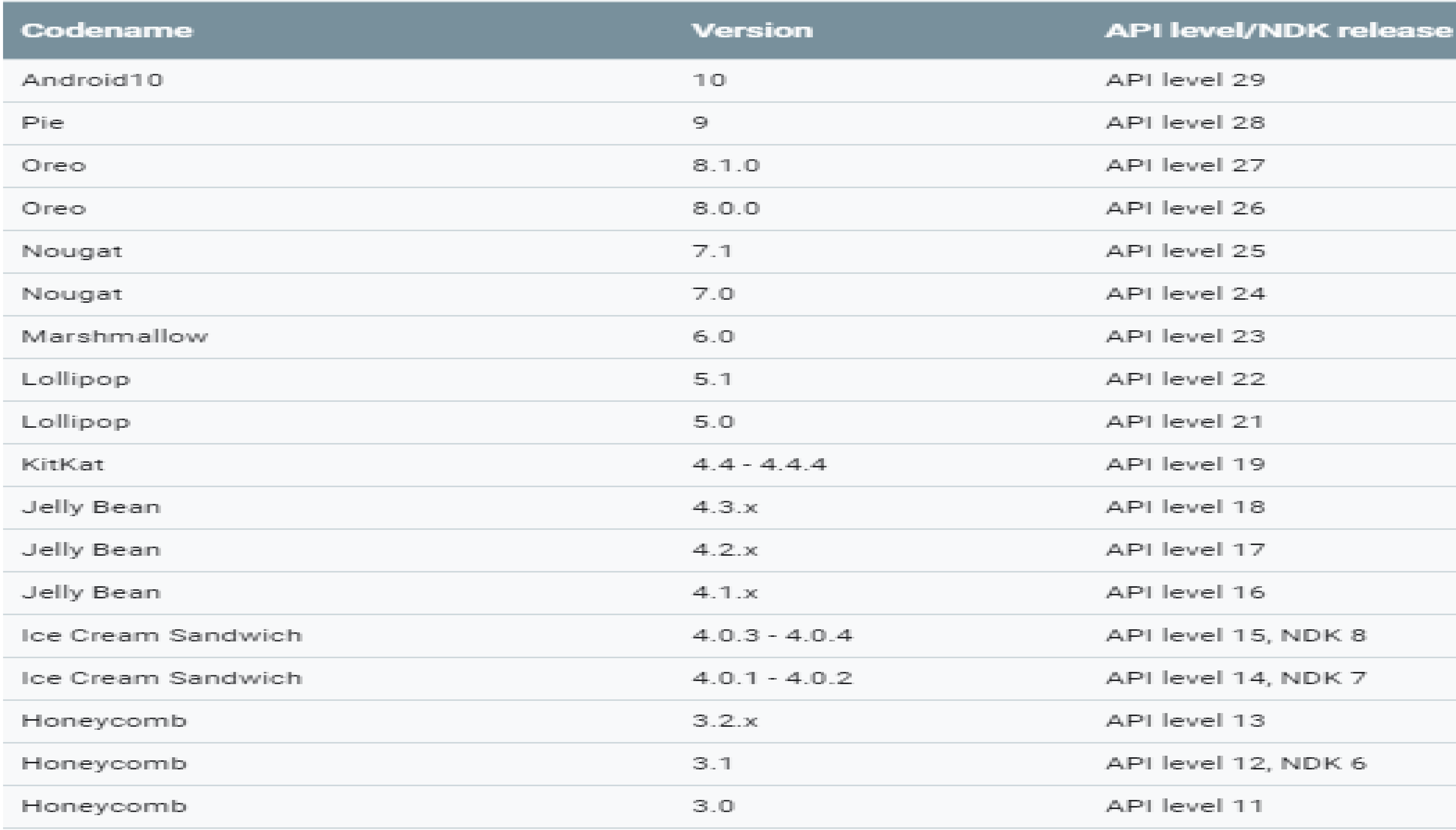
- Complete, open, and free mobile platform with secure OS and robust framework.
- Dalvik Virtual Machine is optimized for mobile devices.
- Telephony support
- Integrated browser based on the open source WebKit engine for rendering web pages
- Media support for common audio, video, and still-image formats (MPEG-4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- Android includes support for high performance 2D and 3D graphics with the Open Graphics Library
- GPS support
- XML support for Designing
- SQLite for structured data storage by default.

Android History and Releases

- Android did not originate with Google. Instead, Android was initially developed by Android, Inc., a small Palo Alto, California–based startup company.
- Google bought this company in the summer of 2005 and released a beta version of the Android Software Development Kit (SDK) in November 2007.
- On September 23, 2008, Google released Android 1.0, whose core features included a web browser, camera support, Google Search, and more.
- Table from the next slide outlines subsequent releases. (Starting with version 1.0, each major release comes under a code name that's based on a dessert item.)
- More Info : https://source.android.com/setup/build-numbers

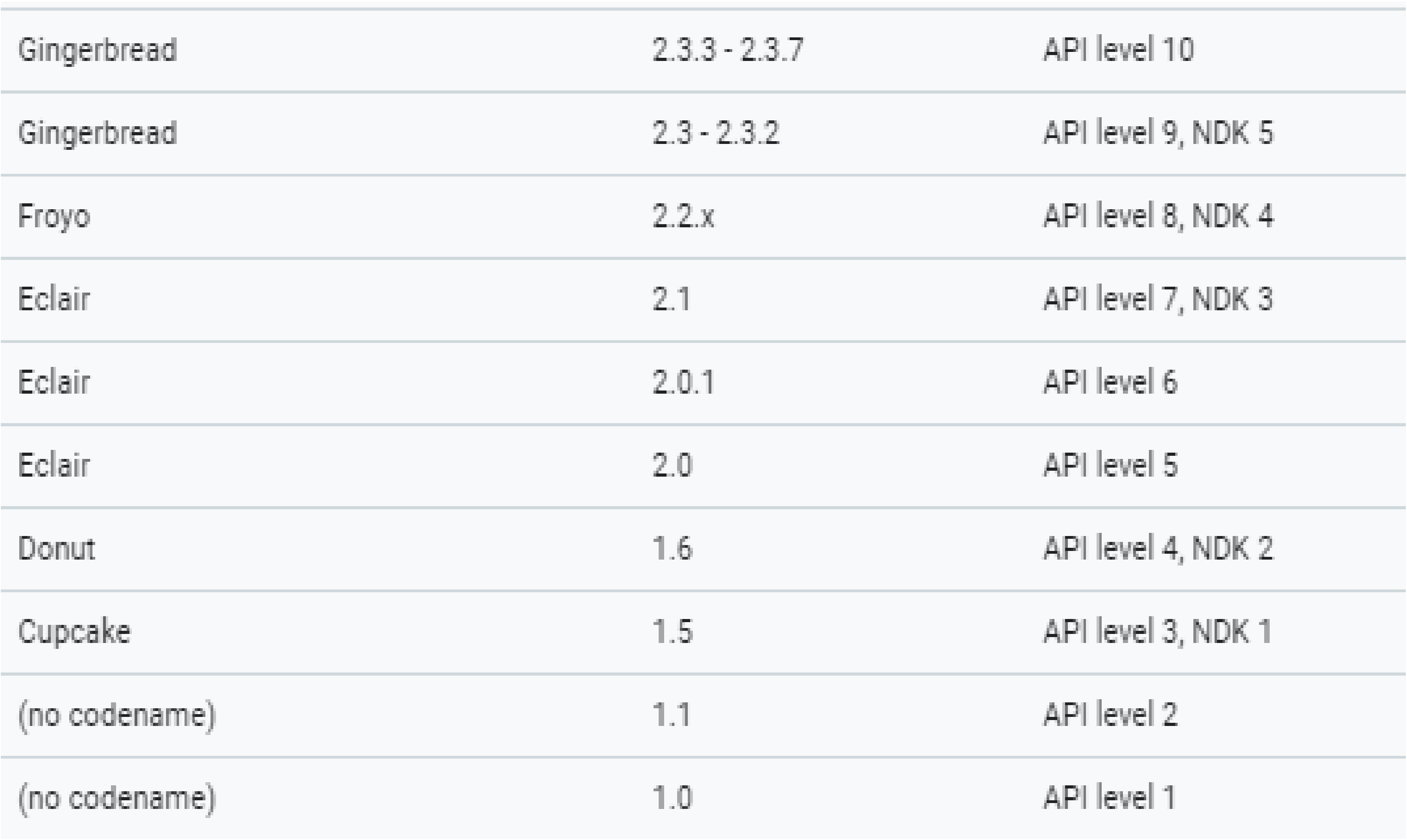
Android Version, Codename, API Level

Ref : https://source.android.com/setup/start/build-numbers



Codename	Version	API level/NDK release
Android10	10	API level 29
Pie	9	API level 28
Oreo	8.1.0	API level 27
Oreo	8.0.0	API level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11

Android Version, Codename, API Level



Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no codename)	1.1	API level 2
(no codename)	1.0	API level 1

Main Point 1

Android programming opens a world of creativity allowing you to express yourself in ways you never dreamed of with the help of its unique architectures and features. *Science of Consciousness: With clear intellectual understanding of higher states of consciousness, students grasp more comprehensive, powerful laws of nature in their own consciousness, resulting in thought, speech and action that express more and more total potential of their creativity.*

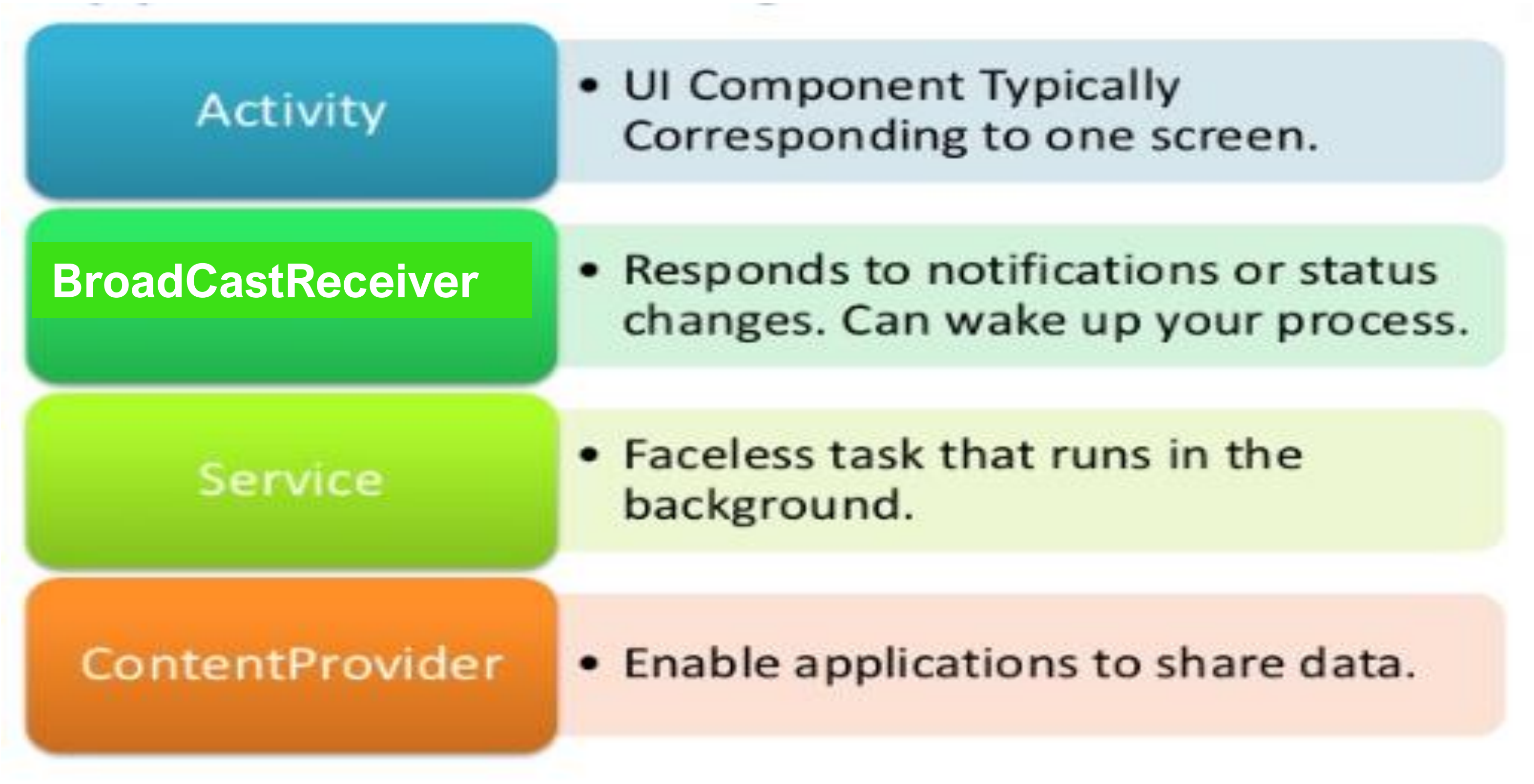
Challenges of Android development

- Multiple screen sizes and resolutions(Solution : Weight Property, Constraint Layout)
- Performance: make your apps responsive and smooth

  (Ref : https://developer.android.com/topic/performance/)
- Security: keep source code and user data safe
  - To make your APK file as small as possible, you should enable shrinking to remove unused code and resources in your release build using Proguard.
  - Ref : https://developer.android.com/studio/build/shrink-code
  - Apply encryption techniques to secure your app data.
- Compatibility: run well on older platform versions.
  - Some versions of android does not support many features. So choose the version used by majority of people.
- Marketing: understand the market and your users
  - Monetize, analyze, and promote your mobile apps with AdMob.

Android Components

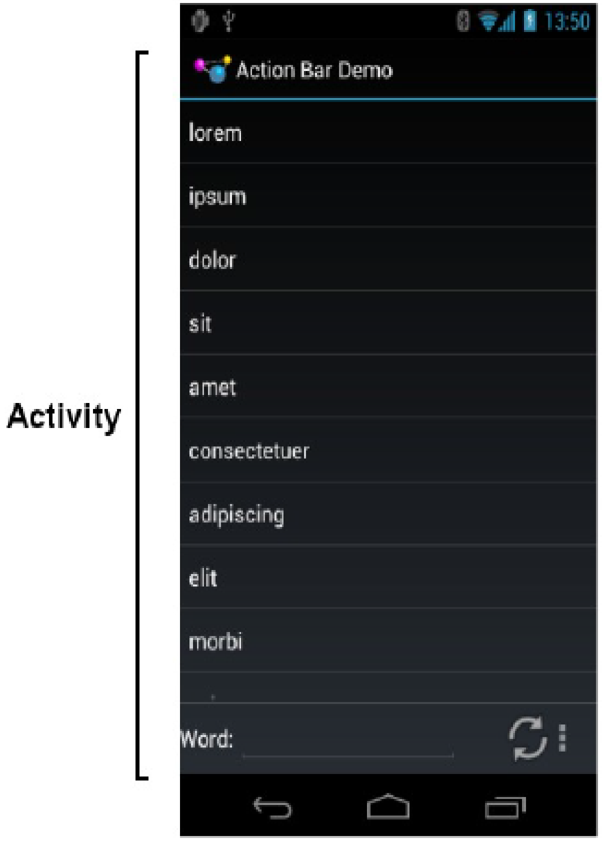
- App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others.



Activity	• UI Component Typically Corresponding to one screen.
BroadCastReceiver	• Responds to notifications or status changes. Can wake up your process.
Service	• Faceless task that runs in the background.
ContentProvider	• Enable applications to share data.

1. Activity

- Every single screen is one activity. It’s a Java or Kotlin file.
- The building block of the user interface is the *activity*.
- It represents a chunk of your user interface and, in some cases, a discrete entry point into your app (i.e., a way for other apps to link to your app).
- When you make an interactive Android program, you start by sub classing the AppCompatActivity
- Activities provide the reusable, interchangeable parts of the flow of UI components across Android applications.



2. Service

- The Android Service class is for background tasks that may be active but not visible on the screen. It works without user interaction.
- Examples
  - When you receive your email updates in inbox it is a service. You get the notification of new e-mail even if you are not using the e-mail app or doing something else.
  - WiFi availability status

3. Content Providers

- Applications can not directly share data between them in Android. It is one of the important security feature. But Content Providers implement a mechanism for the sharing of data between applications.
- A content provider component supplies data from one application to others on request.
- Example: WhatsApp can read data from Contacts with the help of Content Providers.

4. Broadcast Receiver

- The BroadcastReceiver simply respond to broadcast messages from other applications or from the system.
- It is registered for system announcements.
- Broadcast Receivers operate in the background and do not have a user interface.
- Example : Once you insert a headphone, automatically your phone recognize that by showing a headphone symbol, device starts charging, data downloaded.

Quiz

1. Name the top and bottom layers of Android architecture.
2. Which on of the Android app component helps to share the data between different apps?
   a) Activity b) Service c) Content Provider

Main Point 2

- App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others.
  Science of Consciousness: *Regular practice Transcendental Meditation is essential entry point to achieve different aspects of their lives. Life become more orderly, and more rewarding and more successful.*