Android Architecture

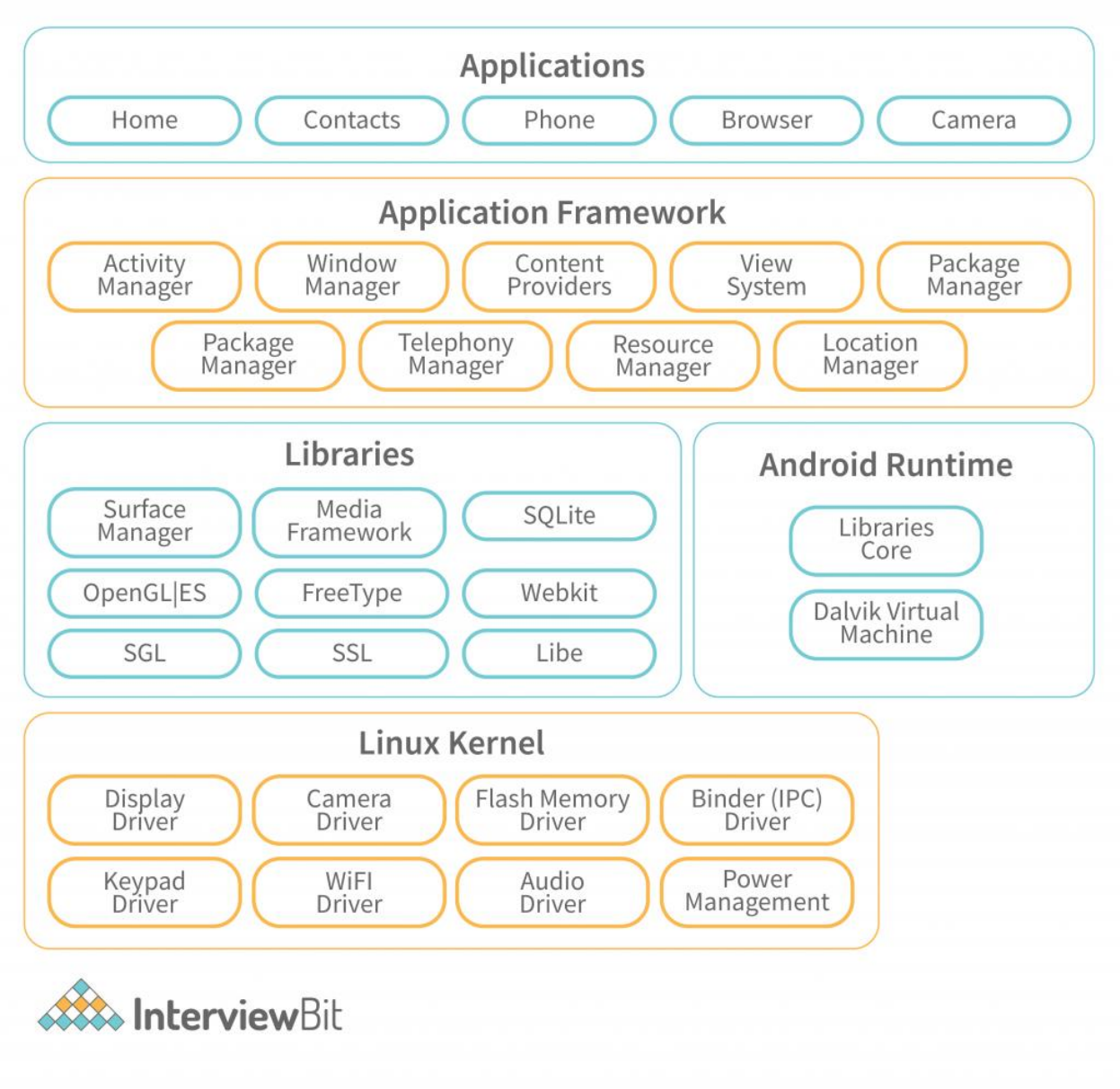
An Android operating system for touchscreen devices poses a number of challenges that must be overcome when developing it. So, in a way, the architecture for an Android platform consists of a set of software components designed to support a large number of Android-enabled devices. This mobile operating system is based on a layered architecture of software stacks comprising a Linux kernel, a runtime environment, supporting libraries, an application framework, as well as a set of applications. As part of building such a complex system, careful structural attention must be paid to make sure all the Android Architect components do not conflict with each other. The Android architecture protects its many components from crashing altogether while keeping their independent functionalities intact. Some of the benefits of Android's layered architecture can be summed up as follows:

- With a layered Android architecture, it will ensure that different problems will be broken down and will be dealt with on different levels.
- Android software developers can avoid low-level problems each time they develop by using a layered architecture. Instead, they can focus on delivering business value relating to the layer they are working on, rather than working on the details.
- There is no need for developers who are working on developing apps to be concerned about the actual implementation of the application framework. Such responsibilities will fall to system developers responsible for implementing the application framework.
- Since Android has a layered structure, it is possible to apply updates incorporating bug fixes or improvements to each layer independently. Keeping changes within layers independent is crucial to ensuring that their effects are not intertwined with one another.
- Developers at different levels of an operating system are able to work together without getting in each other's way. This is particularly important when updating and publishing new versions of an operating system.

Android Components

To support the needs of every Android device, the Android architecture contains a different number of components than the iOS architecture. An Android system can be defined as a set of software stacks consisting of several components. It is important to note that among all the Android Architecture components, Linux Kernel, is responsible for providing the core operating system functions for smartphones, as well as DVM (Dalvik Virtual Machine), which acts as a platform for Android applications to run. The components that make up the android architecture are as follows:

- Linux Kernel
- Platform Libraries
- Android Runtime
- Android Framework
- Applications



Now let's explore each of these in more detail:

1. Linux Kernel

As with any operating system, the Linux kernel, or whatever we call it in our context, is one of the most important components of Android's architecture that resides at the root (bottom layer) of the entire system. It manages all the drivers needed during the runtime of an Android device, such as camera drivers, display drivers, audio drivers, Bluetooth drivers, and memory drivers, among others. Among the main features of the Linux kernel are as follows:

- **Security:** The Linux kernel maintains the security between an application and the host system.

- **Memory Management:** It efficiently manages memory, which allows us to develop our own applications without having to worry about memory allocation.
- **Process Management:** It effectively manages the workflow process and allocates resources when required by processes.
- **Network Stack:** It has the ability to handle network communications effectively and efficiently.
- **Multitasking:** One of the main features of Linux is its support for preemptive multitasking. As a multitasking operating system with asynchronous execution, it enables multiple processes to share the same processors (CPUs) and other resources one at a time. A CPU is dedicated to performing just one task at a time.

Android has evolved a great deal since it was first released and the Linux kernel it runs on has evolved with it too. Below is a table that details the different versions of the Linux kernel.

Android Version	Linux Kernel Version
1.5 Cupcake	(2.6.27)
1.6 Donut	(2.6.29)
2.0/1 Eclair	(2.6.29)
2.2.x Froyo	(2.6.32)
2.3.x Gingerbread	(2.6.35)
3.x.x Honeycomb	(2.6.36)
4.0.x Ice Cream San	(3.0.1)
4.1.x Jelly Bean	(3.0.31)
4.2.x Jelly Bean	(3.4.0)
4.3 Jelly Bean	(3.4.39)
4.4 Kit Kat	(3.10)
5.x Lollipop	(3.16.1)
6.0 Marshmallow	(3.18.10)
7.0 Nougat	3.18.48 4.4.0
7.1 Nougat	
8.0 Oreo	3.18.72 4.4.83 4.9.44
8.1 Oreo	3.18.70 4.4.88 4.9.56
9.0 Pie	4.4.146 4.9.118 4.14.61
10.0 Q	4.9.191 4.14.142 4.19.71

2. Platform/Native Libraries

Several native libraries are layered on top of the Linux kernel. The Library provides the device with a set of instructions that allow it to handle different types of data in an appropriate way. As part of Android development, native libraries are required, and most of these libraries are open-source. It is a collection of C/C++ core libraries as well as Java-based libraries, which support Android development, such as Graphics, Libc, SSL (Secure Socket Layer), SQLite, Media,

Webkit, OpenGL (Open Graphic Library), Surface Manager, etc. Here are some details about some key Android libraries that are available for Android development.

- A media library used for playing, recording, and editing audio and video formats.
- A surface manager library that provides display management functionality.
- A collection of OpenGL (Open Graphic Library) and SGL (Scalable Graphics Library), used for creating 2D and 3D graphics.
- SQLite provides database support, and FreeType provides font support.
- SSL (Secure Socket Layer) provides Internet security and WebKit provides browser support.

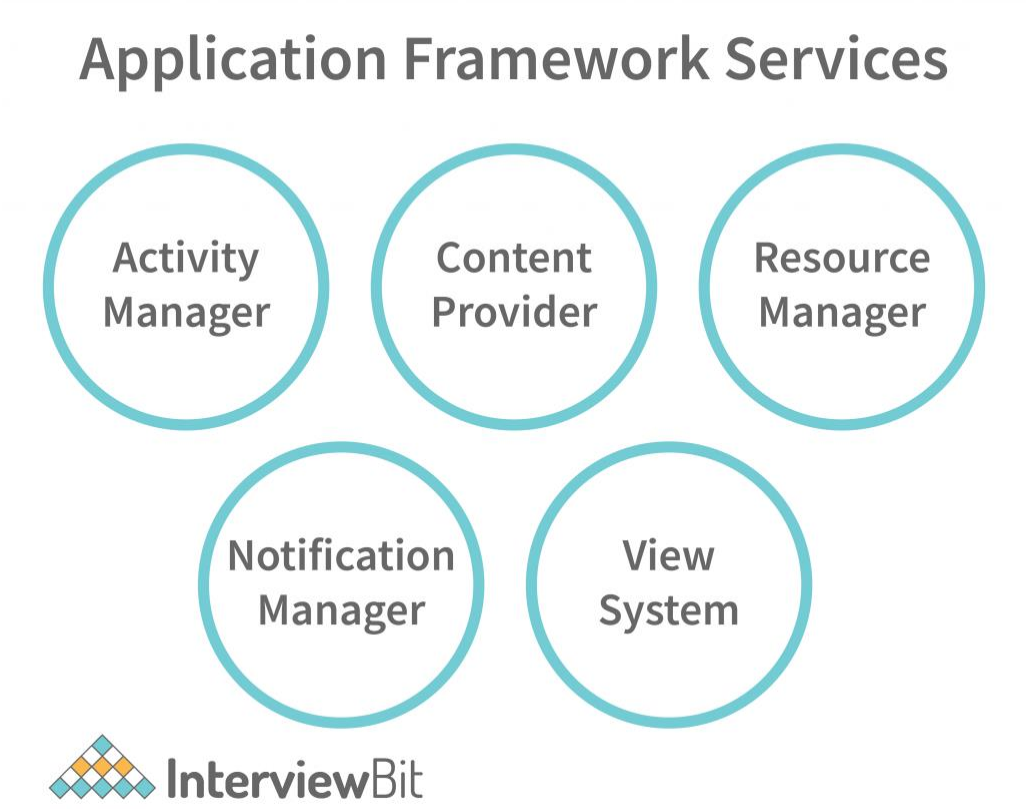
3. Android Runtime

The Android Runtime Environment is an integral part of Android. It contains components such as core libraries and the DVM (Dalvik Virtual Machine). The Android runtime, along with the libraries, powers our applications, and is the basis for the framework.

- Similarly to JVM (Java Virtual Machine), Dalvik Virtual Machine (DVM) is a virtual machine that is used for executing applications on Android. Android utilizes the DVM to optimize its battery life, memory usage, and performance. With its special design and optimization, it can facilitate multiple instances to be run simultaneously on an Android device. In order to handle threading and low-level memory management, it relies on the Linux kernel. This results in faster performance as well as less memory usage.
- The Android runtime provides core libraries which let Android application developers develop Android applications in the Java programming language.

4. Application Framework

The application framework that stands on top of the native libraries and runtime layer provides us with Application Programming Interfaces (APIs) and higher-level services. The Android Application Framework provides classes, interfaces, and utilities that are used for the development of Android applications. Also included in this framework is an Android Hardware Abstraction Layer (HAL), which allows the application to communicate with hardware-specific device drivers and to manage both the UI and resources. The overall aim of it is to provide services through which we can create a particular class and make that particular class available for aid when creating applications. The Android framework includes the following high-level services that can be beneficial for developing mobile applications according to our prerequisites:



- **Activity Manager:** Plays a crucial role in bringing all aspects of the applications lifecycle and activity stack under control.
- **Content Providers:** These providers enable applications to publish data online and share it with other applications.
- **Resource Manager:** Assists in accessing non-code embedded resources such as strings, color settings, and layouts for user interfaces.
- **Notifications Manager:** Allows applications to show alerts and notifications to users via the user interface of the application.
- **View System:** It serves as a base class for widgets and handles events.

5. Applications

Among the layers of the Android architecture, applications are at the top. Applications that come pre-installed on the device, like contacts, music, app store, cameras, galleries, etc., as well as the ones downloaded from the Google Play Store, such as social applications, games, professional applications, etc., will be installed only on this layer. Applications run within Android's runtime environment, utilizing the classes and services provided by the application framework. A typical Android user interacts with this layer mostly for basic functions like accessing the Web browser, making phone calls, viewing galleries, etc. Below are a few examples of some of the standard applications that come pre-installed on every device:

- SMS client app

- Dialer
- Clock
- App store
- Web browser
- Contact manager
- Calculator
- Music, etc.

Conclusion

If you want to lay a good foundation for Android development, one of the first steps should be to get a basic understanding of its overall architecture. Android implementation is carried out by a software stack architecture that includes a Linux kernel, an application framework, libraries, a runtime environment, and a set of applications. Android Studio is a building environment that allows developers to build Android applications in Java or Kotlin and compile them down to bytecode. Upon installation of the application on a device, Android Runtime (ART) compiles the byte code to the native format required by the CPU so that the application can run on the device. A key feature of the Android architecture is its performance and efficiency, both in terms of the performance of applications as well as in the reusability of application design.