Android Software Stack

|  |  |
| --- | --- |
| Application Layer | |
| Application Framework | Core Java |
| Native Libraries | Runtime |
| Linux Kernel | |

Android building blocks

Applications: Android apps are written in Java, using both Java and Android framework libraries.

Application Framework: All things that are needed to write an Android app in Java are found in the framework. They’re comprised of Android classes that handle the UI, resources, how the system operates etc. The framework defines and manages the lifecycles of Android components and how they communicate with one another.

Core Java: These are the core Java libraries used by Android apps and the application framework. It isn’t the full Java Micro Edition or Standard Edition but a subset of them based on the now retired Apache Harmony. Future Android versions after Marshmallow will use the Java libraries from OpenJDK.

Native Libraries: These are C/C++ libraries that deal with media, graphics, databases, OpenGL and so on. Java apps don’t interact directly with native libraries because the application framework has Java wrappers for the native code.

Runtime: This is a self-contained runtime environment that executes Android application code in a virtual machine, which is then translated to bytecode and stored in an executable file. Every app executes its own runtime, Dalvik or the new Android Runtime (ART). The defunct Dalvik, since Lollipop, executable format is meant for systems with processor or memory constraints, which past Android versions fitting the bill. ART’s main improvement is reducing power consumption and executing apps more efficiently. It also has backwards compatibility with Dalvik’s .dex files.

Linux Kernel: This is the foundational operating system that facilitates communication between apps and hardware functionalities like camera, sound, network etc. The kernel manages processes and threads. When an app starts, a process starts with it and every process has a runtime. Inside each process, one or more threads can execute the app code. The kernel has a function called scheduling that schedules how CPU time is allocated for which processes.

Android Application Architecture

Application Object: The android.app.Application object is the representation of the executing Java app. This object is instantiated when the app starts and destroyed when the app closes. A new app object is created with every app restart.

Android Components: The runtime manages components. A few critical components are Activity, Service, BroadcastReceiver and ContentProvider. A component is an entry point where an app can start. An app’s behavior is governed by how the components are configured. A started component can trigger other components that then also trigger others. This chain can last through the entire app lifecycle. A component is started by an intent, which tells it what actions to take. The AndroidManifest.xml file is where all components must be manually registered by the developer.