Android Overview

Table of Contents

[Android Platform Architecture 3](#_Toc100898080)

[Android Features 6](#_Toc100898081)

Long-story short, mobile usage has sky-rocketed in the last decade or so, and a huge contributor to that is the existence of **Android**.

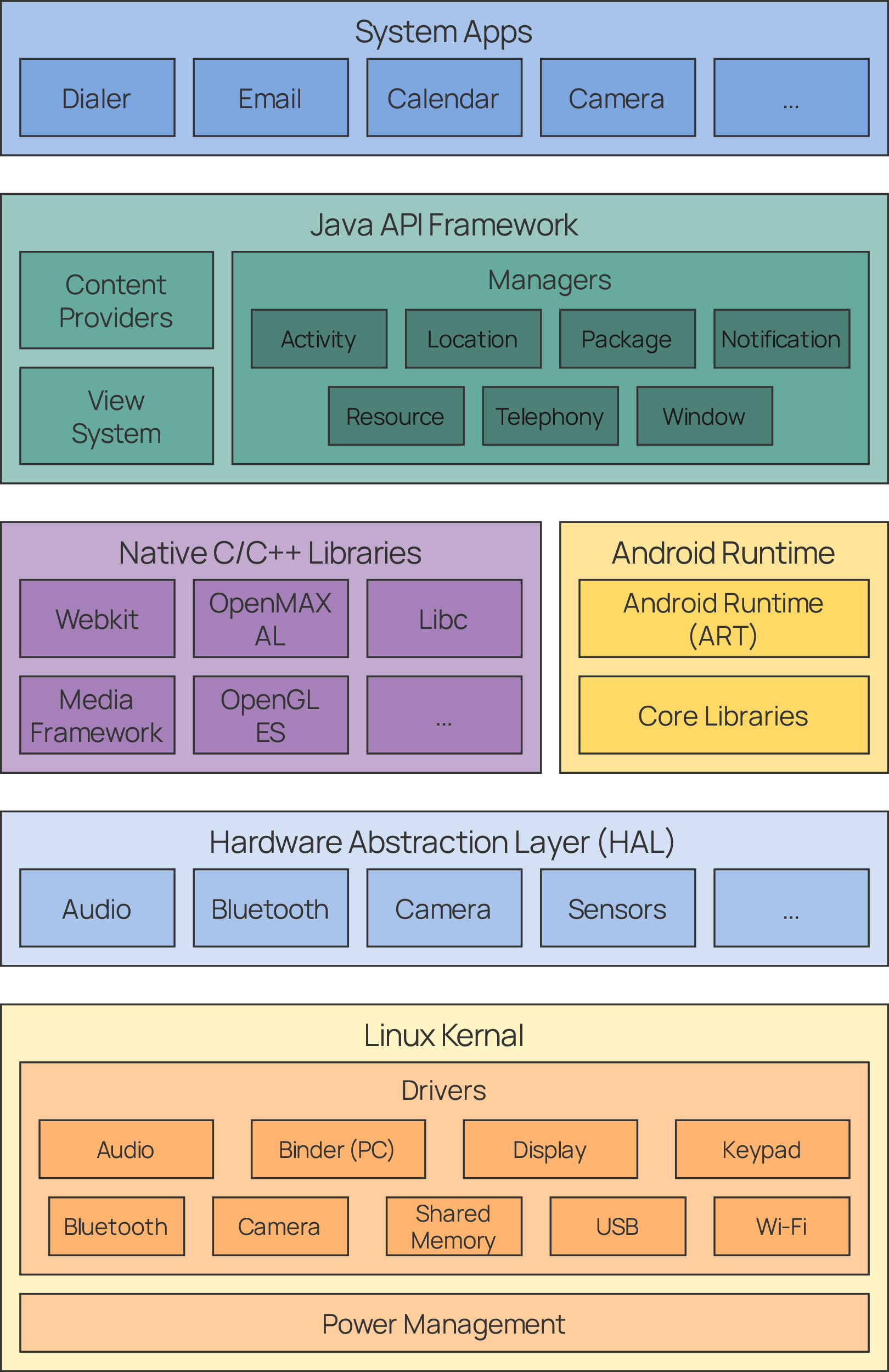
Android is a free, open-source mobile operating system based on Linux. It was originally developed by Android Inc., and was later bought by Google Inc. The Android OS is actually part of a larger software stack that also includes the open-source development platform Android Studio.

The reasons Android is favoured by such a large community include:

* It is a powerful and open-source SDK.
* There are no licensing fees.
* There is a thriving developer community.
* There is a low barrier to entry.
* There is a huge market of users.

## Android Platform Architecture

Like all operating systems, Android is a piece of software that integrates the hardware and the software on the device, essentially directing the software on how to use the hardware.



To achieve its work, Android uses several components:

* **Linux Kernel** – Technically speaking, THIS is the actual OS. Android is a software stack working on top of this. The Linux Kernel takes care of threading and memory management, helping the Android runtime. It also includes several security features like cryptography, rooting, permissions and sandboxing.
* **Hardware Abstraction Layer** (HAL) – This layer provides standard interfaces and drivers, thus allowing access to hardware capabilities to the Java API framework. When the Java API Framework wants to access device hardware, the corresponding library from the HAL is loaded by Android.
* **Android Runtime** (ART) – In the ART layer, build tools are used to compile Java source code into DEX (Dalvik Executable) bytecode, which can run on the Android platform. ART is responsible for compilation, garbage collection, debugging and crash reports. There is also a set of core runtime libraries that provide most of the functionality of the Java language that the Java API Framework uses.
* **Native C/C++ Libraries** – Many core Android system components and services, such as ART and HAL, use native C and C++ libraries. These libraries are also exposed to applications using the Java API Framework and can be accessed using the Android NDK. For example, we can access OpenGL ES to add support for drawing and manipulating 2D and 3D graphics.
* **Java API Framework** – All Android OS features are made available using Java APIs. The APIs simplify the reuse of code, modular system components and services. Developers have access to the same framework APIs that the Android system itself uses. This gives us access to:
  + A rich and extensible view system, which can be used to build the UI
  + A resource manager
  + A notification manager
  + An activity manager, which manages the lifecycle of apps and provides navigation
  + Content providers, which allow the sharing of data with other apps, such as Contacts
* **System Apps** – The built-in applications that come with Android have no special status. This allows users to do things like use a different application as the default browser. Additionally, they provide capabilities to developers. For example, instead of creating our own function to send an SMS, we can invoke the default SMS app to deliver a message to a specified recipient.

More details regarding the different components in the Android platform can be found [here](https://developer.android.com/guide/platform).

## Android Features

* Application framework enabling reuse and replacement of components.
* Integrated browser based on the open source WebKit engine
* Optimized graphics powered by a custom 2D graphics library and 3D graphics based on the OpenGL ES 1.0 specification with optional hardware acceleration
* SQLite support for structured data storage
* Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF, etc.)
* GSM Telephony (hardware dependent)
* Bluetooth, EDGE, 3G, and Wi-Fi (hardware dependent)
* Camera, GPS, compass, and accelerometer (hardware dependent)
* Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE