

# Android Architecture and its components

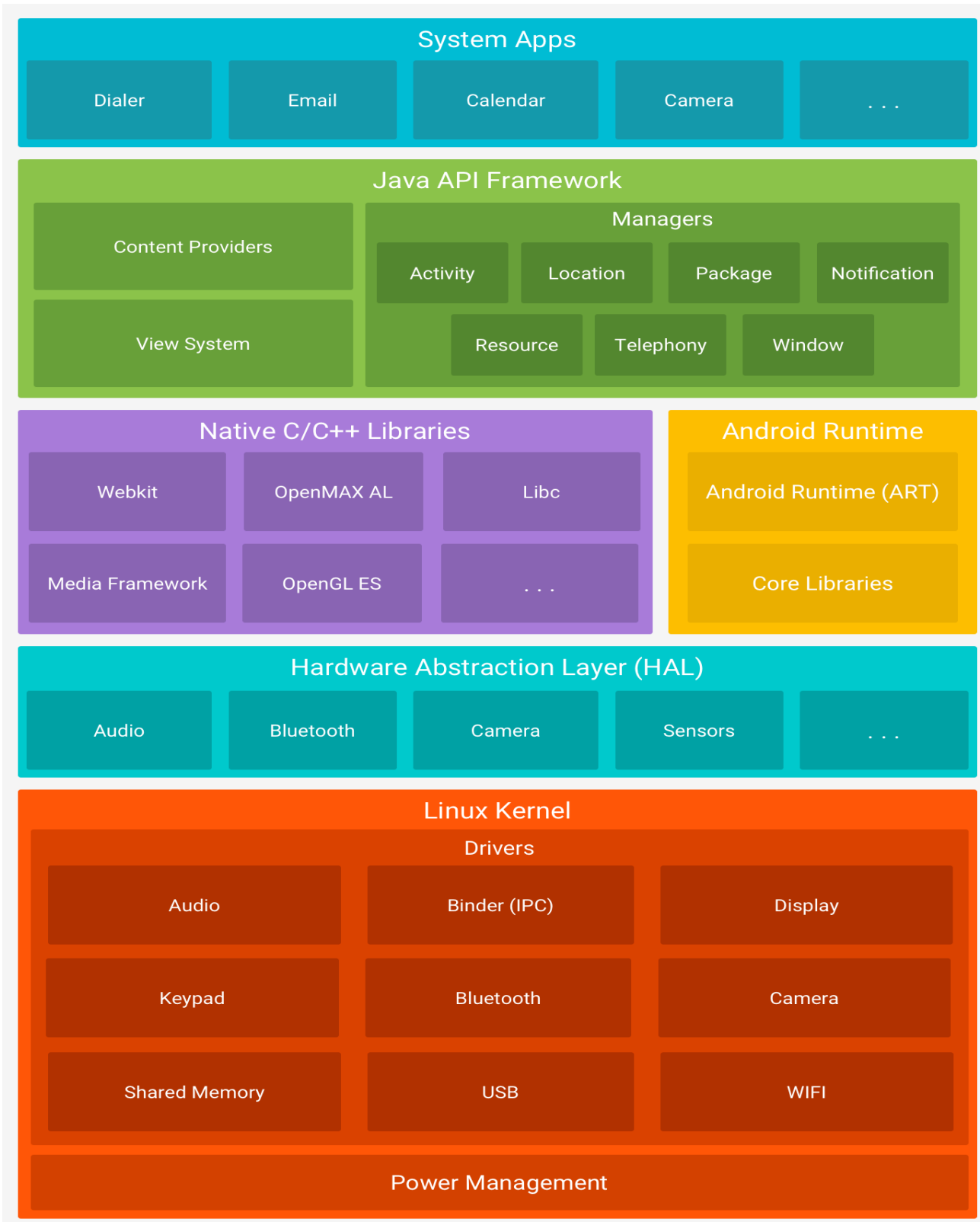
## ➤ What is Android

- Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open-source software and is designed primarily for touchscreen mobile devices such as smartphones and tablets.

## ➤ Key Features of Android OS:

- **Open Source:** Android is an open-source platform, allowing developers to modify and customize the OS according to their needs.
- **App Ecosystem:** Android has a vast ecosystem of applications available through the Google Play Store, providing a wide range of functionality for users.
- **Customization:** Manufacturers can customize the Android OS to create their own versions, resulting in a variety of Android devices with different user interfaces and features.
- **Android Development:** Android development refers to the process of creating applications (apps) that run on the Android operating system. Developers use programming languages like Java or Kotlin, and they typically use the Android Studio Integrated Development Environment (IDE) for creating Android applications.
  - Android apps are primarily written in Java or Kotlin.
  - Android Studio is the official IDE for Android development.
  - The Android Software Development Kit (SDK) provides tools and libraries necessary for developing Android apps.

## ➤ Android Platform Architecture



**Android architecture is designed in a layered fashion, with each layer providing specific functionalities. The key components of Android architecture include:**

### **1. Linux Kernel:**

The Linux kernel serves as the foundation of the Android operating system. It provides core system services, such as security, memory management, process management, and device drivers.

The Linux kernel in Android is responsible for managing hardware resources like the CPU, memory, and device drivers for components like camera, display, etc.

### **2. Hardware Abstraction Layer (HAL):**

The hardware abstraction layer (HAL) provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware component, such as the camera or Bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component.

If an Android device has a specific camera sensor, the HAL provides a standardized interface that allows the rest of the system to access the camera without knowing the details of that particular sensor.

### **3. Native Libraries:**

Many core Android system components and services, such as ART and HAL, are built from native code that requires native libraries written in C and C++. The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps. For example, you can access OpenGL ES through the Android framework's Java OpenGL API to add support for drawing and manipulating 2D and 3D graphics in your app. If you are developing an app that requires C or C++ code, you can use the Android NDK to access some of these native platform libraries directly from your native code.

### **4. Android Runtime:**

Android Runtime (ART) is the managed runtime used by the Android operating system for executing and managing Android applications. It replaced the earlier Dalvik runtime, starting with Android 5.0 (Lollipop). The transition from Dalvik to ART brought significant

improvements in performance, memory efficiency, and overall responsiveness of Android devices.

For devices running Android version 5.0 (API level 21) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing Dalvik Executable format (DEX) files, a bytecode format designed specifically for Android that's optimized for a minimal memory footprint. Build tools, such as d8, compile Java sources into DEX bytecode, which can run on the Android platform.

**Some of the major features of ART include the following:**

- Ahead-of-time (AOT) and just-in-time (JIT) compilation
- Optimized garbage collection (GC)
- On Android 9 (API level 28) and higher, conversion of an app package's DEX files to more compact machine code
- Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields

Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime. If your app runs well on ART, then it can work on Dalvik as well, but the reverse might not be true.

Android also includes a set of core runtime libraries that provide most of the functionality of the Java programming language, including some Java 8 language features, that the Java API framework uses.

## **5. Java API framework:**

The entire feature-set of the Android OS is available to you through APIs written in the Java language. These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components and services, which include the following:

- A rich and extensible view system you can use to build an app's UI, including lists, grids, text boxes, buttons, and even an embeddable web browser
- A resource manager, providing access to non-code resources such as localized strings, graphics, and layout files
- A notification manager that enables all apps to display custom alerts in the status bar

- An activity manager that manages the lifecycle of apps and provides a common navigation back stack
- Content providers that enable apps to access data from other apps, such as the Contacts app, or to share their own data

Developers have full access to the same framework APIs that Android system apps use.

## **6. System apps:**

Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install. So, a third-party app can become the user's default web browser, SMS messenger, or even the default keyboard. Some exceptions apply, such as the system's Settings app.

The system apps function both as apps for users and to provide key capabilities that developers can access from their own app. For example, if you want your app to deliver SMS messages, you don't need to build that functionality yourself. You can instead invoke whichever SMS app is already installed to deliver a message to the recipient you specify.