Mobile App Development - Android

Introduction to Android

Jigisha Patel

Agenda

- Introduction to Android Platform
- Overview of Android architecture

What is Android ?

android

android

- Google
- Open Source
- Mobile Device OS
- Linux Kernel

Kotlin

Java

Android Version History

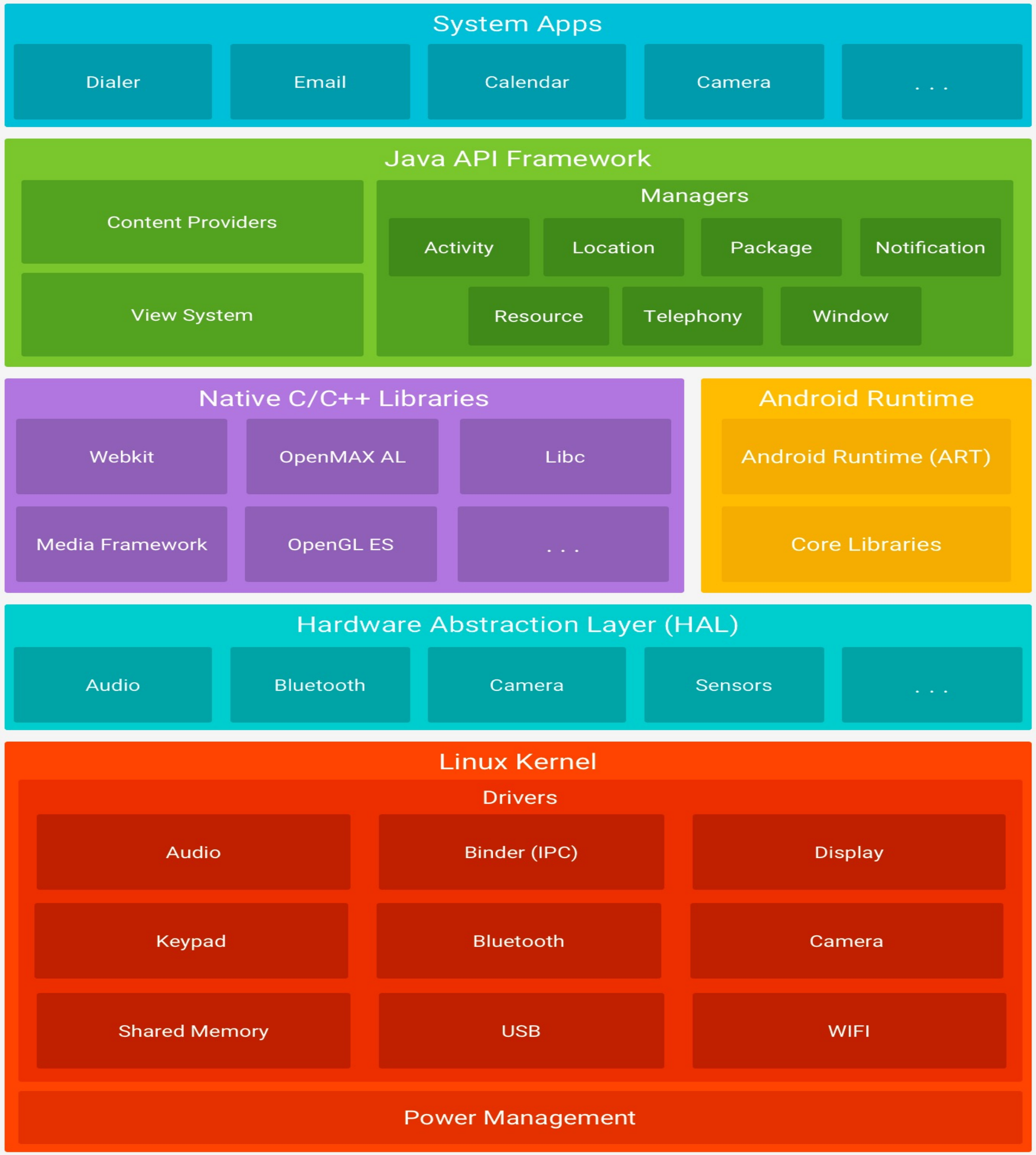
Codename	Version	API level/NDK release
Android 11	11	API level 30
Android 10	10	API level 29
Pie	9	API level 28
Oreo	8.1.0	API level 27
Oreo	8.0.0	API level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
… Marshmallow, Lollipop, KitKat, JellyBean, IceCream Sandwich, Honeycomb, Froyo, Éclair, Donut, Cupcake		

What's New in Android 11 ?

Android Development Requirements

- Android Studio
- Android Virtual Device (AVD)
- Android SDK

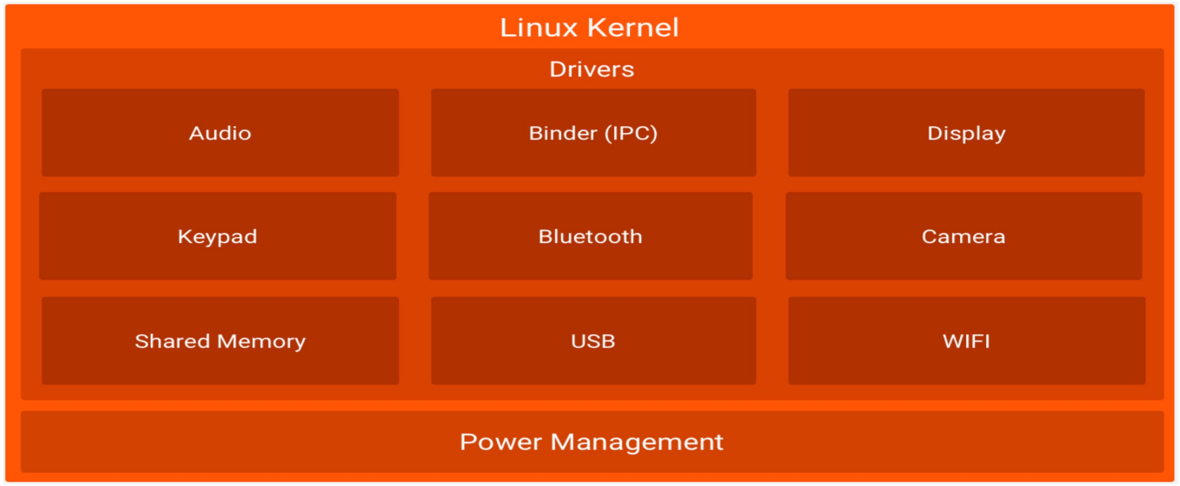
Android Platform Architecture



- Image source: https://developer.android.com/guide/platform

The Linux kernel

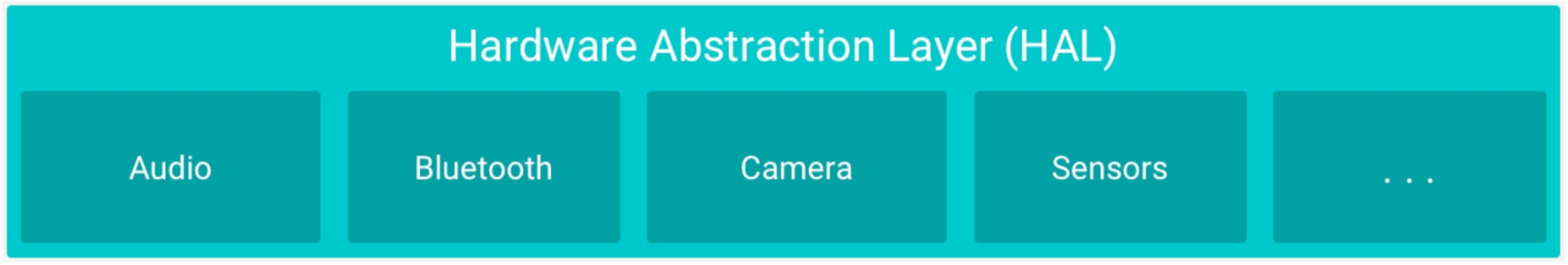
- The foundation of the Android platform is the Linux kernel.
- For example, the Android Runtime (ART) relies on the Linux kernel for underlying functionalities such as threading and low-level memory management.
- Using a Linux kernel allows Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.



source: https://developer.android.com/guide/platform

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.

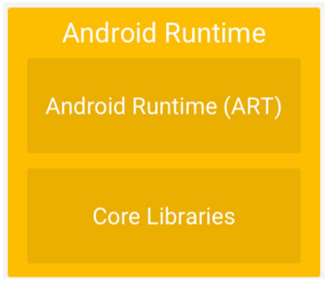


source: https://developer.android.com/guide/platform

Android Runtime (ART)

- 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).

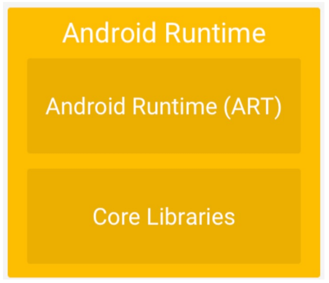
- ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed specially for Android that's optimized for minimal memory footprint.



source: https://developer.android.com/guide/platform

Android Runtime (ART) cont…

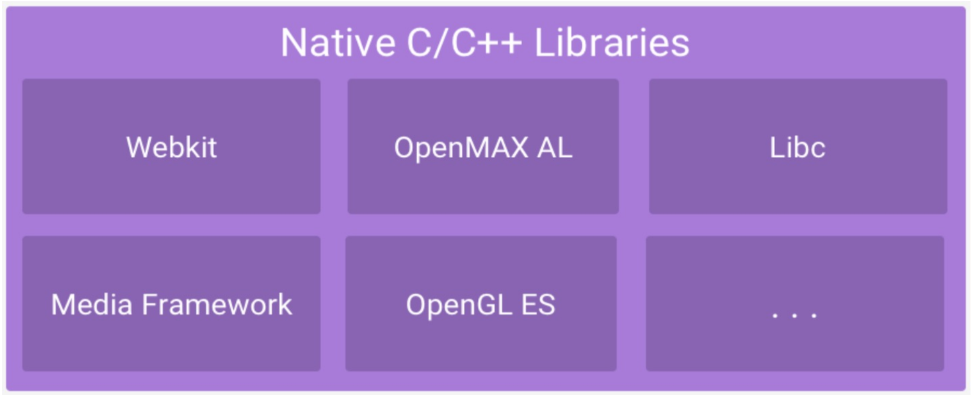
- Some of the major features of ART include the following:
  - Ahead-of-time (AOT) and just-in-time (JIT) compilation
  - Optimized garbage collection (GC)
  - On Android 9 (API level 28) and higher, conversion of an app package's Dalvik Executable format (DEX) files to more compact machine code.
  -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



source: https://developer.android.com/guide/platform

Native C/C++ Libraries

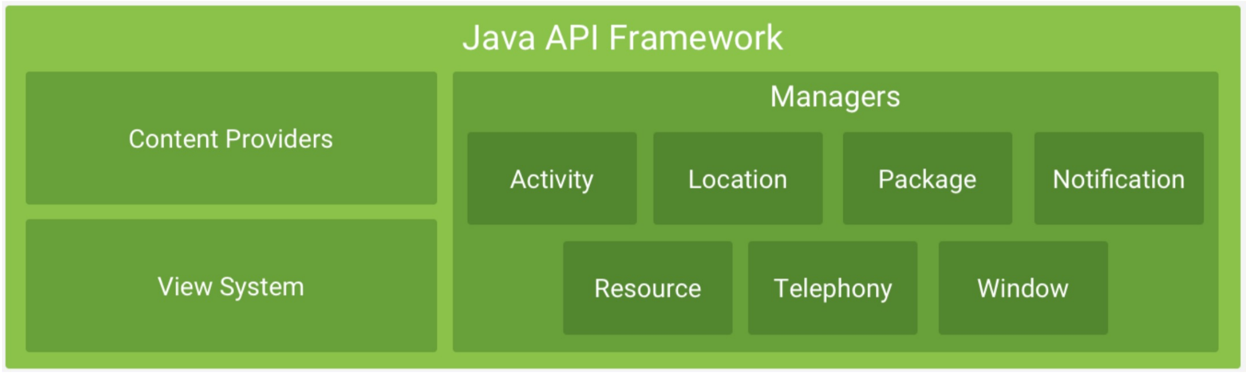
- Many core Android system components and services, such as ART and HAL, are built from native code that require native libraries written in C and C++.
- For example, you can access OpenGL ES through the Android framework's Java OpenGL API to add support for drawing and manipulating 2D and 3D graphics 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.



source: https://developer.android.com/guide/platform

Java API Framework

- The entire feature-set of the Android OS is available to you through APIs written in the Java language.
- Developers have full access to the same framework APIs that Android system apps use.
- These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components and services.



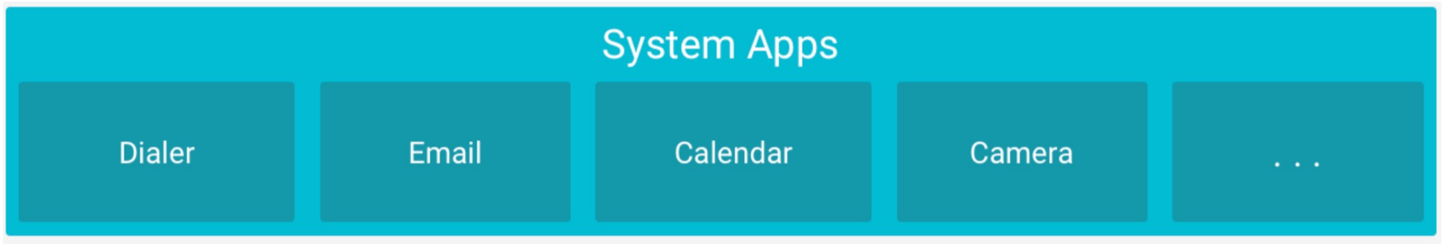
source: https://developer.android.com/guide/platform

Java API Framework cont…

- Services provided by Java API Framework:
  - A rich and extensible View System you can use to build an app’s UI
  - 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

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.



source: https://developer.android.com/guide/platform

Gradle

- Gradle is an open-source build automation tool focused on flexibility and performance.
- Gradle build scripts are written using a Groovy or Kotlin DSL.
- **Highly customizable**
  - Gradle is modeled in a way that is customizable and extensible in the most fundamental ways.
- **Fast**
  - Gradle completes tasks quickly by reusing outputs from previous executions, processing only inputs that changed, and executing tasks in parallel.
- **Powerful**
  - Gradle is the official build tool for Android, and comes with support for many popular languages and technologies.

References

- Android Programming with Kotlin for beginners by Horton, J.
- https://developer.android.com/guide/platform