Building Software Systems

Lecture 4.3

Introduction to Android App Development

SAURABH SRIVASTAVA

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IIT (ISM) DHANBAD

What is Android?

Android is a free, open-source operating system designed primarily for mobile devices

- It is built on top of the Linux kernel and includes other open-source software
- Android powers billions of smartphones, tablets, and other devices today
- Since it is open-source, it allows developers and manufacturers to modify and distribute their own versions
- These are colloquially known as “custom ROMs”, e.g., Samsung One UI, Xiaomi MIUI, OnePlus OxygenOS etc.

Google leads the development and maintenance of Android

- It includes overseeing the direction of the platform, releasing updates, and enforcing security practices
- While Android is open-source under the Android Open Source Project (AOSP), Google manages this project
- Google offers a suite of APIs and system services through Google Play Services
- Google Play Store is the most prominent marketplace for Android apps
- Play Store provides tools for developers to publish, update, and manage their apps
- Various measures to ensure the security and privacy of apps distributed through the Play Store are in place

Android Architecture

Linux Kernel:

- The foundation of the Android platform, handling hardware abstraction and device drivers

Hardware Abstraction Layer (HAL):

- Provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework

Android Runtime (ART):

- Includes the core libraries and serves as the runtime environment for Android applications

Native Libraries:

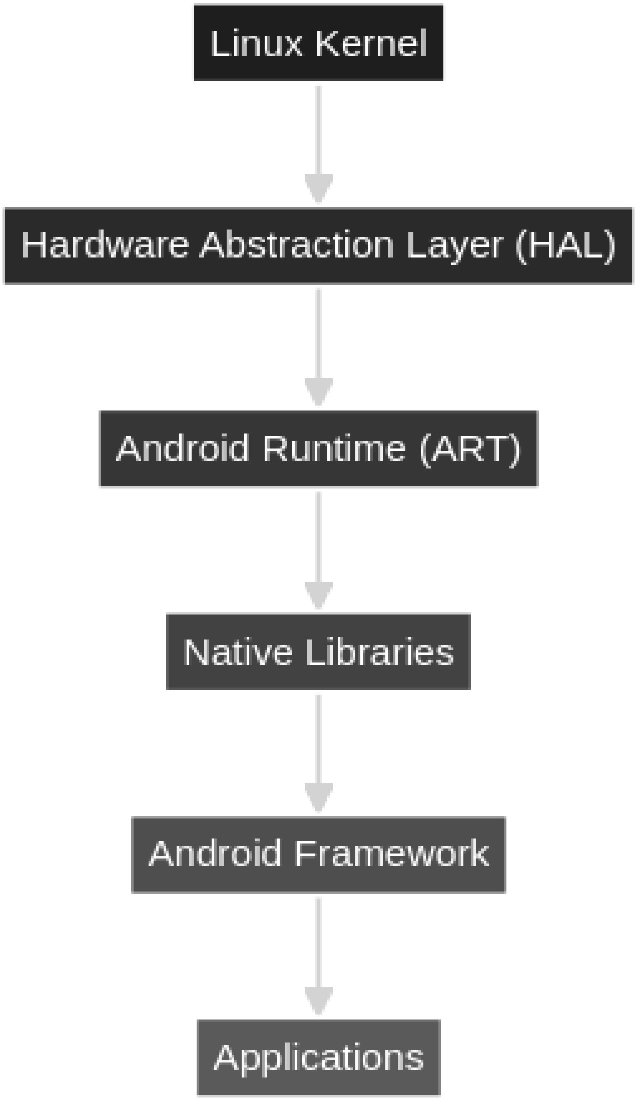
- C/C++ libraries used by various components of the Android system

Android Framework:

- Provides the APIs necessary to build Android applications

Applications:

- The top layer where all Android applications are located, both system and third-party apps



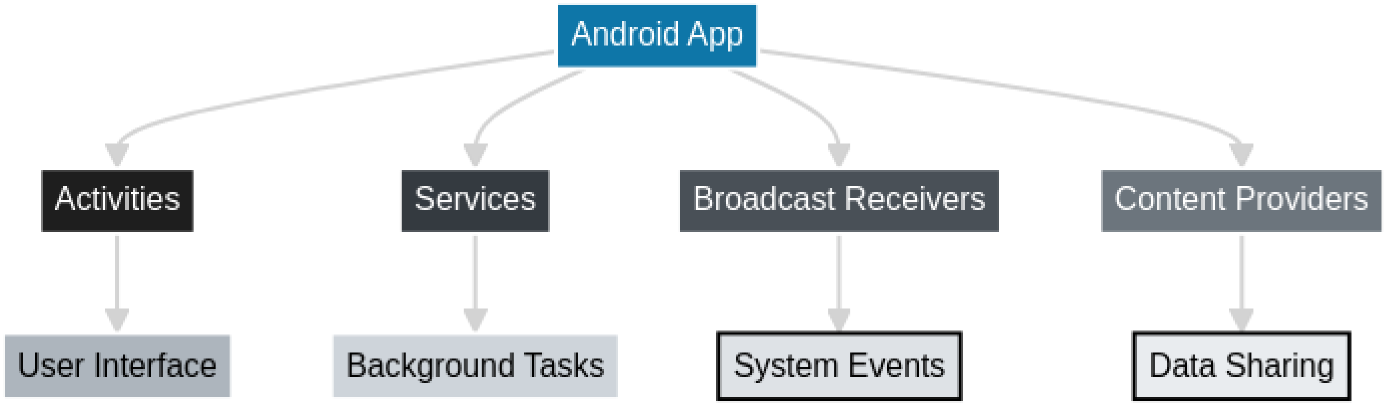
Logical View of a typical Android App

Activities handle user interactions and UI

Services manage background tasks without a UI

Broadcast Receivers respond to system-wide broadcast announcements

Content Providers offer a way to share data between applications



Hybrid vs Native Apps

Native Apps are developed for a specific platform (Android or iOS)

- They use platform-specific programming languages (Java/Kotlin for Android, Swift/Objective-C for iOS)
- They can directly access device hardware and software

Hybrid Apps are developed using web technologies (HTML, CSS, JavaScript)

- They are then wrapped in a native container
- They run on a web view and can work across multiple platforms

Development Languages and Tools

- Native: Platform-specific languages and tools (Android Studio, Xcode)
- Hybrid: Web technologies (HTML, CSS, JavaScript), plus frameworks like Ionic or Cordova

Performance

- Native: Generally, offer superior performance and smooth user experience
- Hybrid: Can experience performance issues, especially with complex animations or graphics