



Mobile Application Development Platforms

**IT6306 - Mobile Application
Development**

Level III - Semester 5

Overview

- The focus is on two main mobile application development platforms, Android and iOS.
- The architectures of the two platforms will be discussed and more emphasis is given on Android platform.
- Finally, points out facts to consider when choosing the right platform for the development of a mobile application.

Intended Learning Outcomes

At the end of this lesson, you will be able to;

- Describe different mobile application platforms
 - Android
 - iOS
- Explain Android platform's architecture
- Explain iOS architecture
- Compare Android and iOS mobile application platforms
- Describe the factors affecting the decision of choosing the right platform.

List of sub topics





- 2.1. Operating systems for mobile applications
- 2.2. Android development platform
- 2.3. iOS development platform
- 2.4. Selecting a development platform

2.1 Operating Systems for Mobile Applications




There are different mobile OS platforms. For example,

- Android (Google)
- iOS (Apple)
- Bada (Samsung)
- Blackberry OS (Research in Motion)
- Windows OS (Microsoft)
- Symbian OS (Nokia)
- Tizen (Samsung)

Operating Systems Contd..

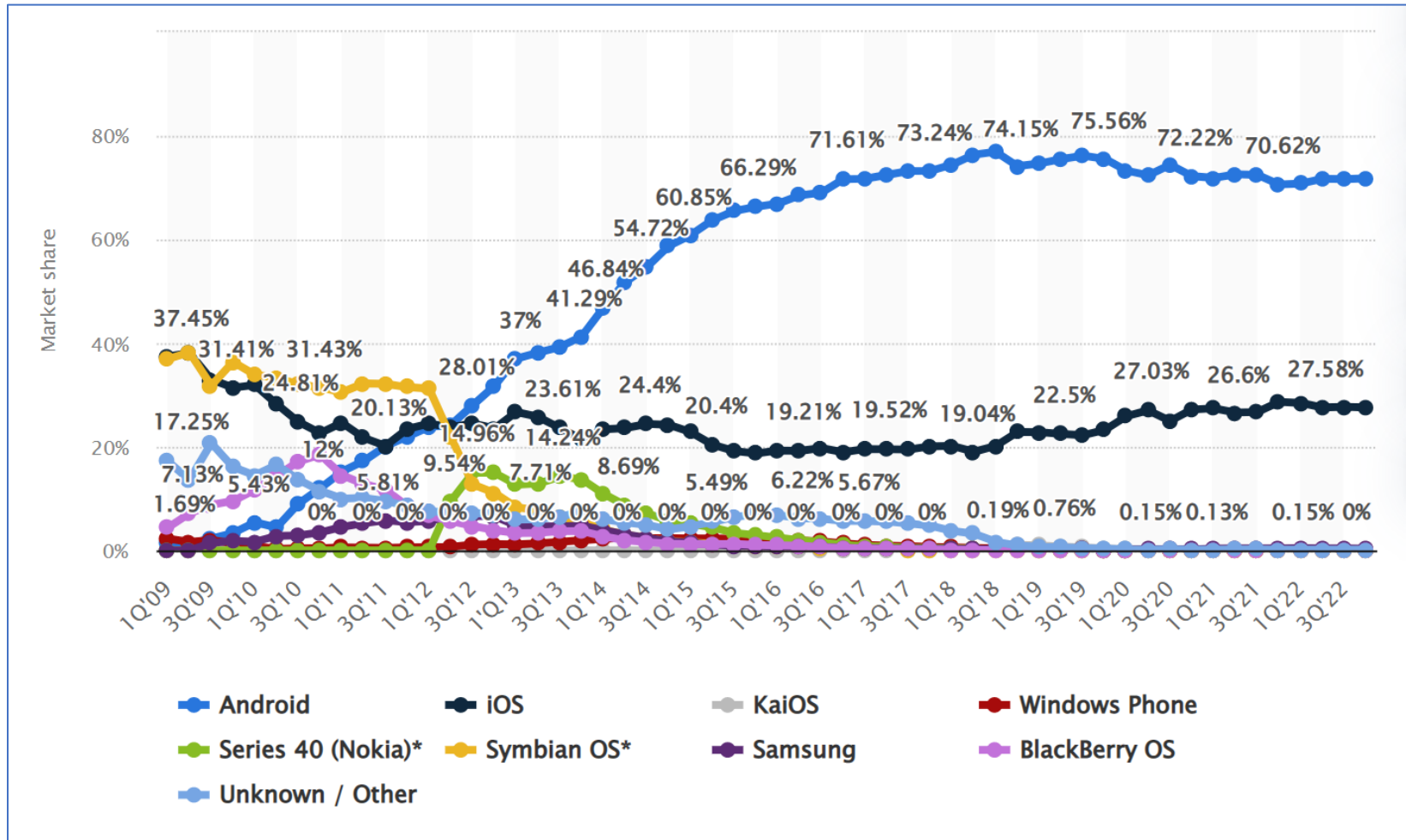
OS	Description	Latest Version
Android	The Android operating system is the most popular operating system today. It is based on the Linux Kernel and open-source software. The android operating system was developed by Google.	
Bada	Bada is a Samsung mobile operating system that was launched in 2010.	
BlackBerry	The BlackBerry operating system is developed by Research In Motion (RIM). It was designed specifically for BlackBerry handheld devices.	
iPhone OS	The iOS was developed by the Apple inc. It is a very secure operating system.	

Operating Systems Contd..

OS	Description	Icon
Symbian OS	The Symbian operating system is based on the java language. Nokia was the first company to release Symbian OS on its mobile phones	
Window mobile OS	The window mobile OS is developed by Microsoft. It was designed for the pocket PCs and smart mobiles.	
Web OS	The WebOS is a mobile operating system that was developed by Palm. It based on the Linux Kernel.	

Mobile OS Platforms Market Share (2009 – 2022)

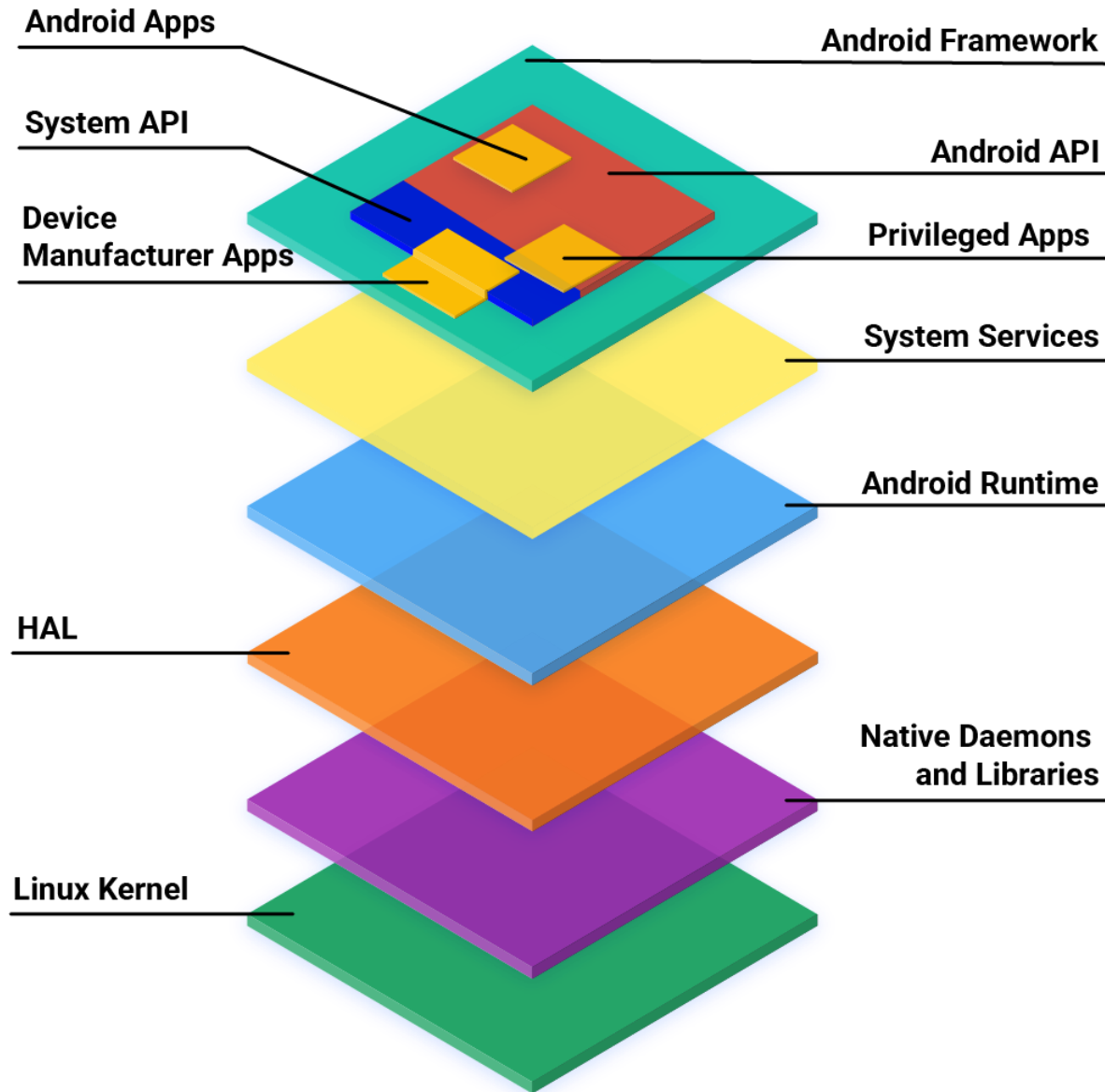
- Android, iOS and Windows became leaders over the time.



2.2 Android Development Platform

- The *Android Open System Platform (AOSP)* is publicly available and modifiable Android source code.
- Anyone can download and modify AOSP for their device.
- AOSP provides a complete and fully functional implementation of the Android mobile platform.
- There are two levels of compatibility for devices implementing AOSP:
 - AOSP compatibility
 - Android compatibility

AOSP Architecture



Android Open System Platform Architecture

The software stack for AOSP contains several layers:

Android app layer

- Android API is used to create apps.
- Apps can be uploaded to and downloaded from the Google Play Store.

Privileged app

- An app created using a combination of the Android and system APIs.
- These apps must be preinstalled as privileged apps on a device.

Android Platform

Device manufacturer app layer

- An app created using a combination of the Android API, system API, and direct access to the Android framework implementation.
- Device manufacturer might directly access unstable APIs
- Such apps must be preinstalled on the device
- These apps are updated with the system software update

System API

- The System API represents Android APIs available only to partners and Original Equipment Manufacturers (OEMs)
- These APIs are marked with `@SystemApi` annotation in the source code.

Android Platform

Android API

- The Android API is the publicly available API for third-party Android app developers.

Android Framework

- A group of Java classes, interfaces, and other precompiled code used to build apps.
- Portions of the framework are publicly available
- Other portions of the framework are available only to OEMs

Android Platform

System Services

- System services are modular, focused components
e.g. `system_server`, `SurfaceFlinger`, and `MediaService`
- Functionality exposed by Android framework API communicates with system services to access the underlying hardware.

Android Runtime (ART)

- A Java runtime environment provided by AOSP.
- ART performs the translation of the app's bytecode into processor-specific instructions
- And the instructions are executed by the device's runtime environment.

Android Platform

Hardware Abstraction Layer (HAL)

- HAL is an abstraction layer with a standard interface
- Using a HAL lets you implement functionality without affecting or modifying the higher level system

Native Daemons and Libraries

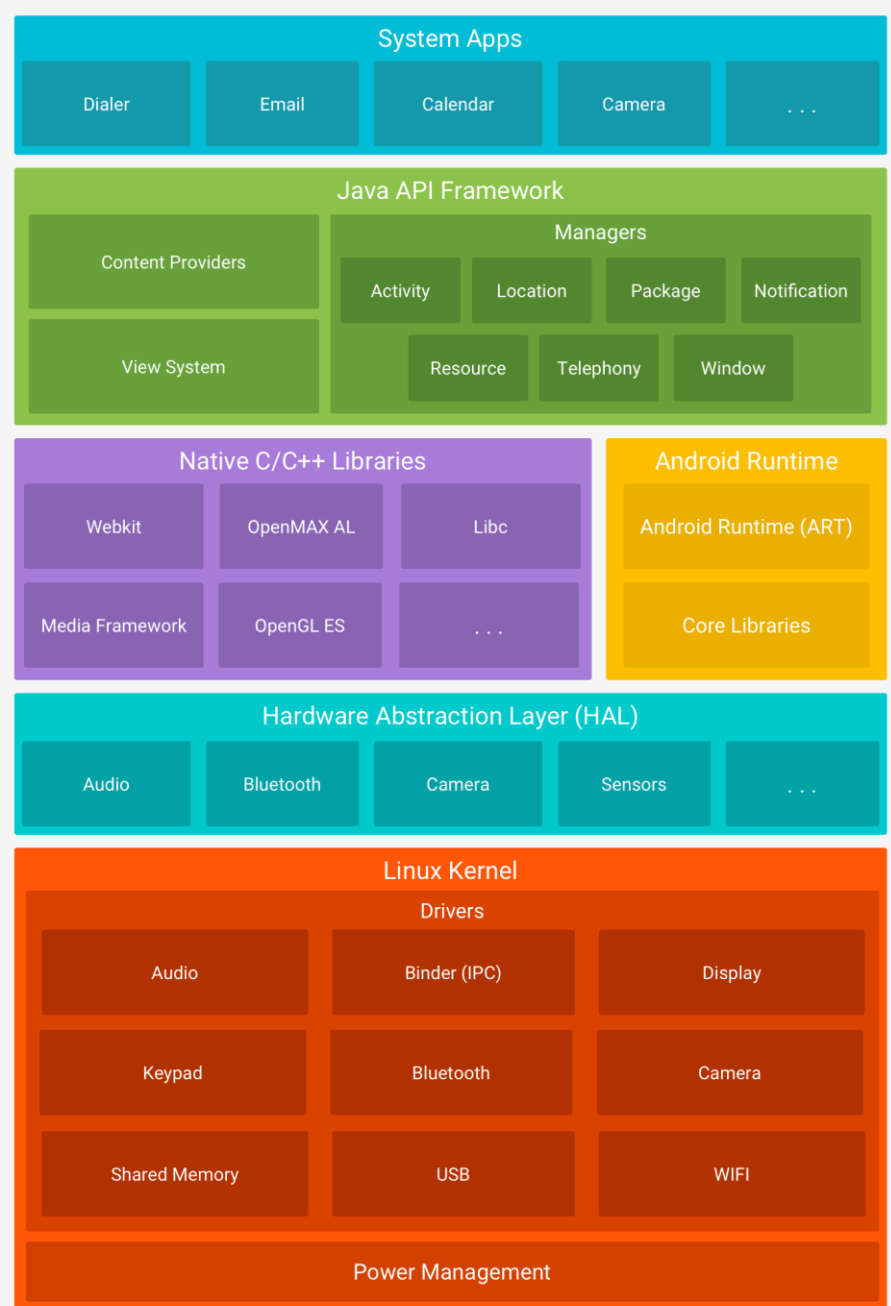
- Native daemons in this layer include init, healthd, logd, and stored
- These daemons interact directly with the kernel or other interfaces

Kernel

- The kernel is the central part of any operating system
- It is used to communicate to the underlying hardware on a device

Android Software Stack Components

- Android is an open source, Linux-based software stack
- It supports a wide array of devices and form factors
- This shows the major software components of the Android platform



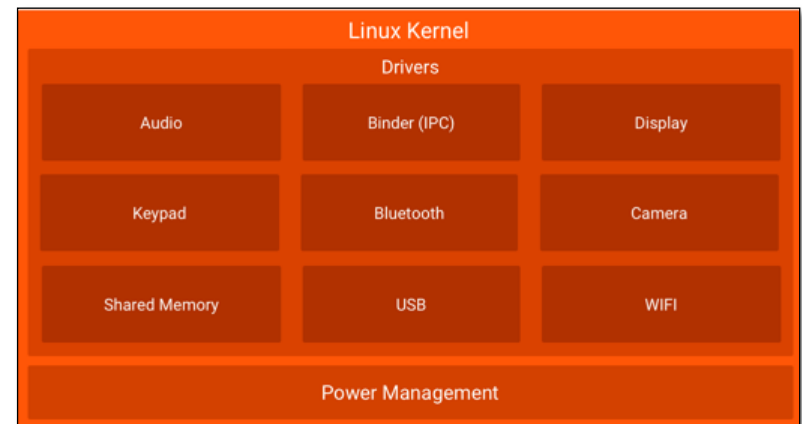
Source: <https://developer.android.com/guide/platform>

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 (Threading and memory management)

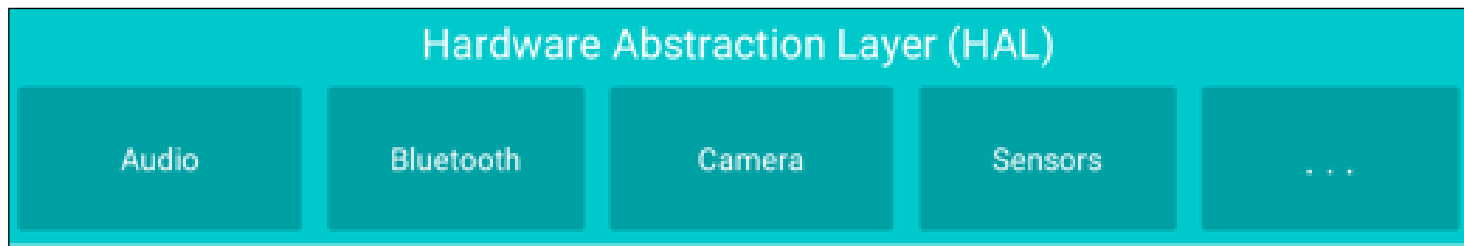
- Using a Linux kernel allows Android to take advantage of key security features
- This allows device manufacturers
- to develop hardware drivers for a
- well-known kernel



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.
- It has multiple library modules, each of which implements an interface for a specific type of hardware component (e.g. 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.



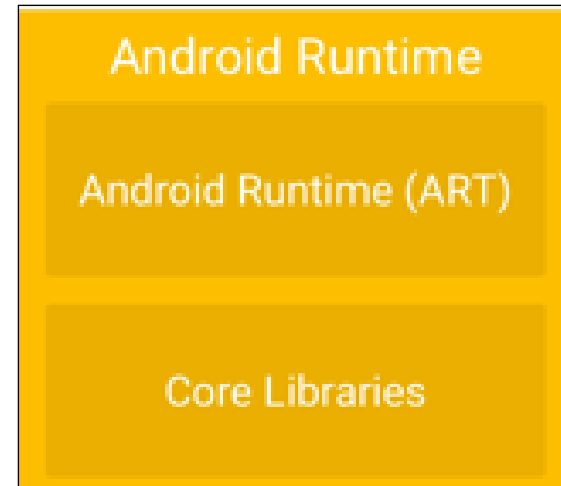
Source: <https://developer.android.com/guide/platform>

Android Runtime

- Android Runtime (ART) is an application runtime environment used by the Android operating system
- ART performs the translation of the application's bytecode into native instructions
- 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

DEX file:

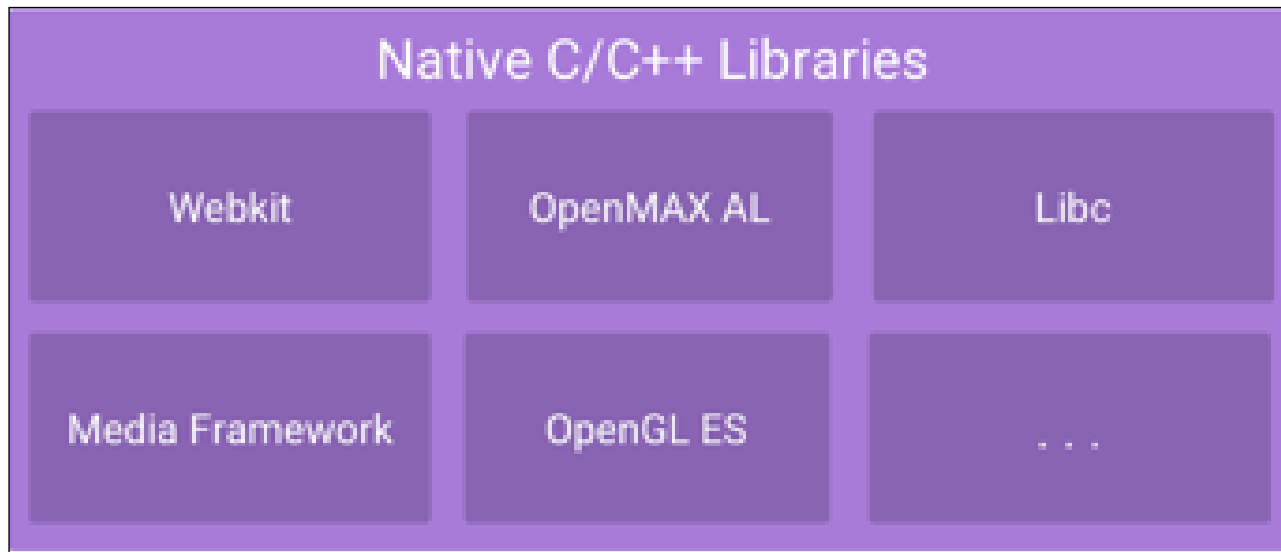
“a bytecode format designed specially for Android and is optimized for minimal memory footprint”



Source: <https://developer.android.com/guide/platform>

Native C/C++ Libraries

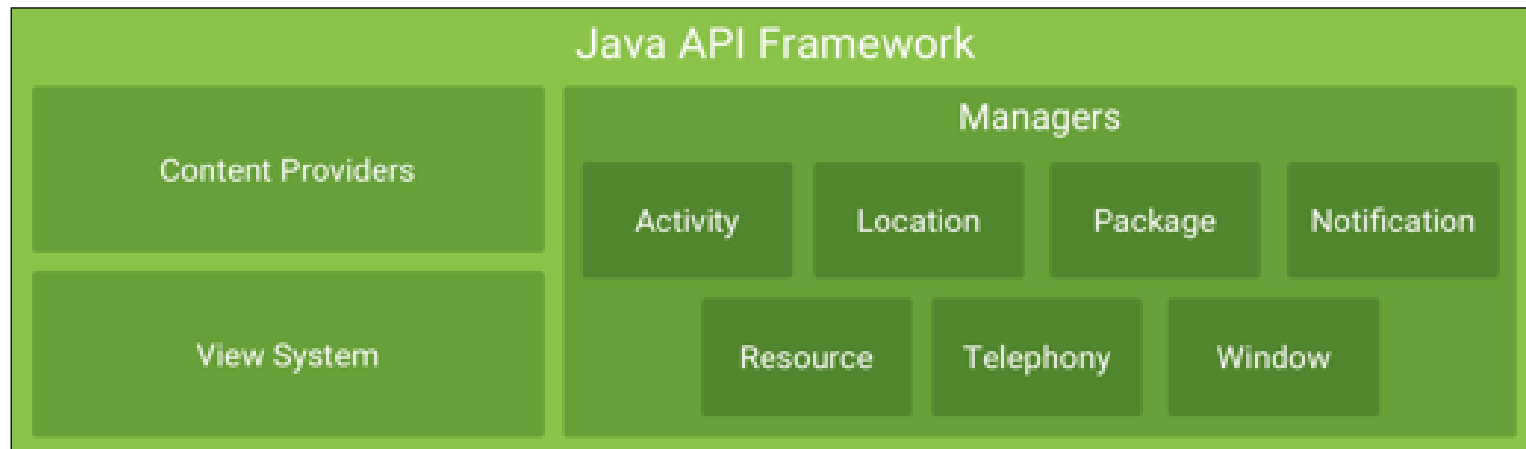
- ART and HAL, are built from native code that require native libraries written in C and C++
- The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps
- For example, you can access OpenGL ES through the Android framework's Java OpenGL API to add support for drawing and manipulating 2D and 3D graphics in your app



Source: <https://developer.android.com/guide/platform>

Java API Framework

- Android OS is available 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.



Source: <https://developer.android.com/guide/platform>

Java API Framework contd..

System components and services include,

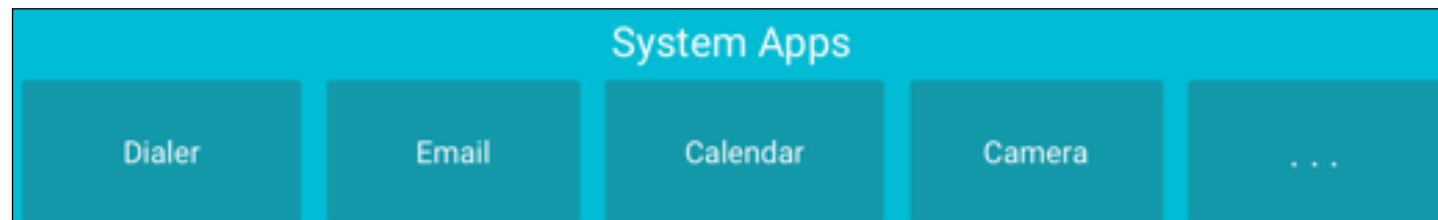
- A rich and extensible View System to build an app's UI
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
- A Notification Manager that enables all apps to display custom alerts in the status bar
- An Activity Manager that manages the lifecycle of apps and provides a common navigation back stack
- Content Providers that enable apps to access data from other apps

System Apps

- Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, etc..
- 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 your app would like to deliver an SMS message, you don't need to build that functionality yourself
- So invoke whichever SMS app is already installed to deliver a message to the recipient



Source: <https://developer.android.com/guide/platform>

Android Devices

- Beyond phones and tablets, the Android OS serves as a platform for a diverse array of hardware and technology

For example,

- automotive (motor vehicle) interfaces
- unified corporate networks
- entertainment devices, etc..

2.3 iPhone Operating System (iOS) Development Platform

- iOS is the proprietary mobile operating system of Apple Inc.
- It supports following programming languages
 - Objective-C
 - C
 - C++
 - Swift
- It is based on the Macintosh OS X
- World's second most popular mobile operating system
- iOS offers user-centric designs



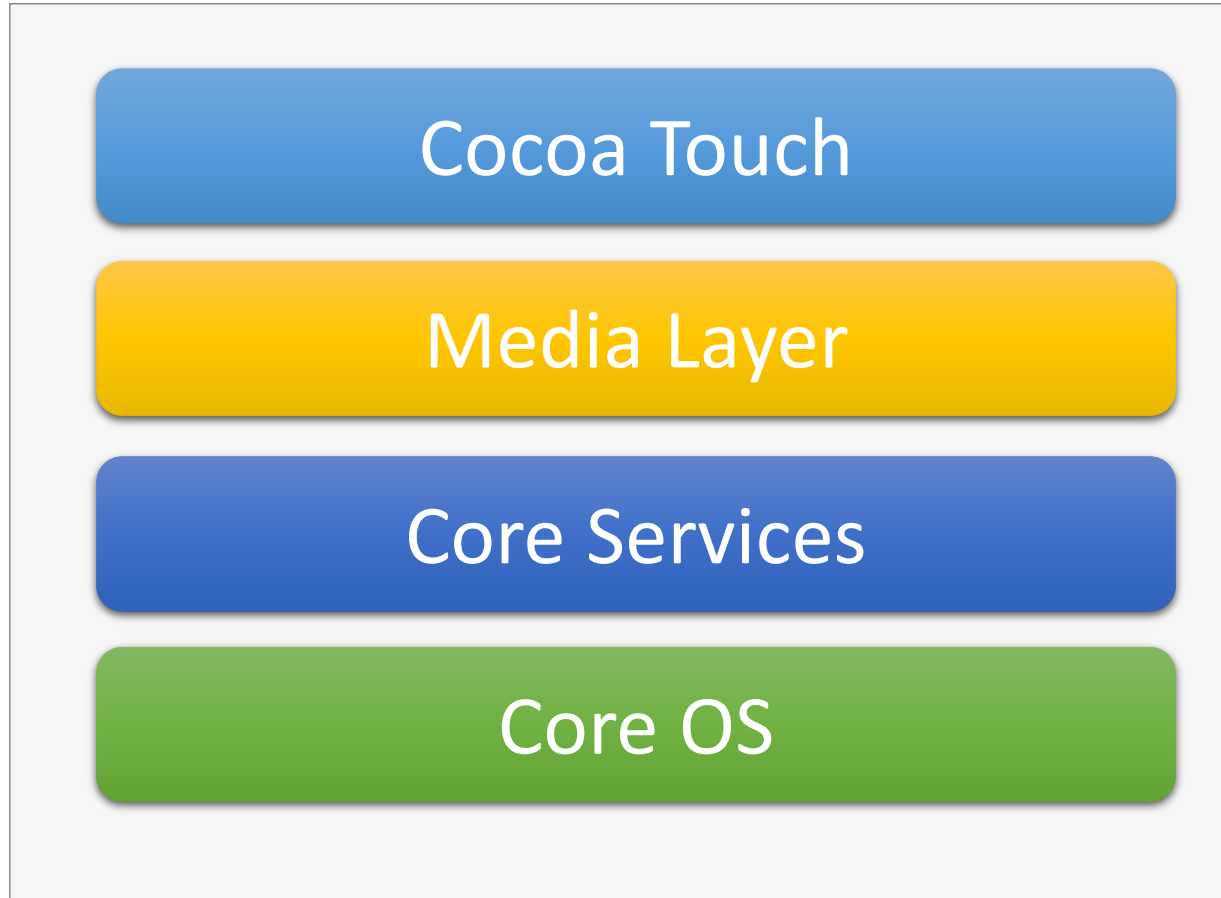
iOS Version History

- The iPhone was first released in June 2007
- In June 2010 Apple rebranded iPhone OS as iOS

iPhone OS 1	1.1.5	July 15, 2008
iPhone OS 2	2.2.1	January 27, 2009
iPhone OS 3	3.1.3	February 2, 2010
	3.2.2	August 11, 2010
iOS 4	4.2.1	November 22, 2010
	4.2.10	July 25, 2011
	4.3.5	
iOS 5	5.1.1	May 7, 2012
iOS 6	6.1.6	February 21, 2014
iOS 7	7.1.2	June 30, 2014
iOS 8	8.4.1	August 13, 2015
iOS 9	9.3.5	August 25, 2016
	9.3.6	July 22, 2019

iOS 9	9.3.5	August 25, 2016
	9.3.6	July 22, 2019
iOS 10	10.3.3	July 19, 2017
	10.3.4	July 22, 2019
iOS 11	11.4.1	July 9, 2018
iOS 12	12.5.7	January 23, 2023
iOS 13 / iPadOS 13	13.7	September 1, 2020
iOS 14 / iPadOS 14	14.8.1	October 26, 2021
iOS 15 / iPadOS 15	15.7.3	January 23, 2023
iOS 16 / iPadOS 16	16.3	

iOS Layered Architecture



Core OS Layer

- Core Bluetooth Framework
- Accelerate Framework
- External Accessory Framework
- Security Services framework
- Local Authentication framework

Core Services Layer

- **Address book framework**
 - Provides programmatic access to a contacts database of user.
- **Cloud Kit framework**
 - Medium for moving data between your app and iCloud.
- **Core data Framework**
 - Technology for managing the data model of a Model View Controller app.
- **Core Foundation framework**
 - Interfaces that gives fundamental data management and service features for iOS apps

Core Services Layer Contd..

- **Core Location framework**
 - Gives location and heading information to apps
- **Core Motion Framework**
 - Using this core motion framework Accelerometer based information can be accessed
- **Foundation Framework**
 - Objective C covering too many of the features found in the Core Foundation framework

Core Services Layer Contd..

- **Healthkit framework**
 - Handles health-related information of users
- **Homekit framework**
 - Communicating with and controlling connected devices in a user's home
- **Social framework**
 - Simple interface for accessing the user's social media accounts
- **StoreKit framework**
 - Gives support for the buying of content and services from inside your iOS apps, a feature known as In-App Purchase

Media Layer

- **UIKit Graphics**
 - support for designing images and used for animating the content of your views
- **Core Graphics framework**
 - the native drawing engine for iOS apps. It supports for custom 2D vector and image-based rendering.
- **Core Animation**
 - optimizes the animation experience of your apps.
- **Core Images**
 - support for controlling video and motionless images in a nondestructive way

Media Layer Contd..

- **OpenGL ES and GLKit**
 - manages advanced 2D and 3D rendering by hardware accelerated interfaces
- **Metal**
 - It permits very high performance for your sophisticated graphics rendering and computation works

Media Layer Contd..

Audio Framework:

- **Media Player Framework**
 - gives simple use to a user's iTunes library and support for playing playlists
- **AV Foundation**
 - it is an Objective C interface. It handles recording and playback of audio and video
- **OpenAL**
 - an industry standard technology for providing audio

Media Layer Contd..

Video Framework:

- **AV Kit**
 - provides interfaces for presenting video
- **AV Foundation**
 - gives advanced video playback and recording capability
- **Core Media**
 - describes the low level interfaces and data types for operating media

Cocoa Touch Layer

- **EventKit framework**
 - provides view controllers for showing the standard system interfaces and enables seeing and altering calendar related events
- **GameKit Framework**
 - allows users share their game related information online
- **iAd Framework**
 - deliver banner-based advertisements
- **MapKit Framework**
 - provides a scrollable map

Cocoa Touch Layer Contd..

- **PushKitFramework**
 - provides registration support for VoIP apps
- **Twitter Framework**
 - generating tweets and support for creating URLs to access the Twitter service
- **UIKit Framework**
 - provides infrastructure for graphical, event-driven apps

2.4 Selecting the Proper Development Platform

- Selecting the right platform among the available is important
- If you do not choose the correct platform, it affects the overall success of the applications developed



Factors to Pay Attention

- **Target Audience of the Mobile Application**
 - Research about the possible audience, their device preferences and key interests
- **Cost involved with the development**
 - Need to do a proper cost estimation. There may be a significant difference in costs associated with different platforms
- **Devices support you expect**
 - How many devices should be supported by your application?
 - Eg: iOS will only support a limited number of devices

Factors to Pay Attention Contd..

- **Application business model**
 - Determine the application's clear intent; monetization options, and the relationship between the application and users.
 - For example, is it for business, entertainment, charity, etc..
- **Technical features**
 - It is important to assess how easy to implement the technical features required.
 - The platform chosen should support the required features easily
- **Third-party integrations**
 - Platforms should make it easier to integrate confidentially with third-party software

Summary

Mobile OSs

Different mobile OSs and market capitalization.

Android Platform

Explains the Android Operating system architecture and version history.

iOS Platform

Explains the iOS Operating system and its components.

Which Platform to Choose

Factors to consider when selecting a Mobile Application Development platform (OS)

Additional References

Android platform architecture - Software Stack
[<https://developer.android.com/guide/platform>]

Android Core [<https://source.android.com/docs/core>]