


Architecture of IOS Operating System

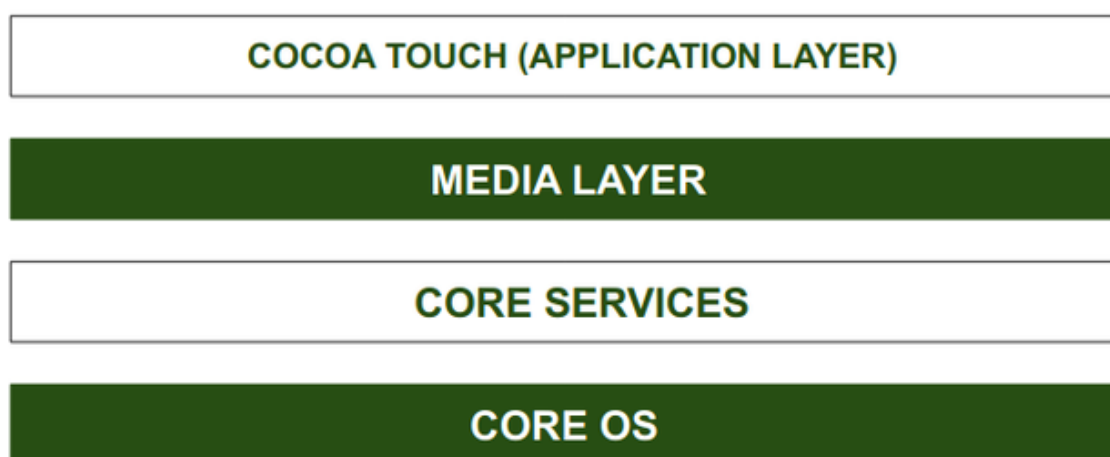
 [geeksforgeeks.org/architecture-of-ios-operating-system/amp/](https://www.geeksforgeeks.org/architecture-of-ios-operating-system/amp/)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

iOS is a Mobile Operating System that was developed by Apple Inc. for iPhones, iPads, and other Apple mobile devices. iOS is the second most popular and most used Mobile Operating System after Android.

The structure of the iOS operating System is Layered based. Its communication doesn't occur directly. The layer's between the Application Layer and the Hardware layer will help for Communication. The lower level gives basic services on which all applications rely and the higher-level layers provide graphics and interface-related services. Most of the system interfaces come with a special package called a framework.

A framework is a directory that holds dynamic shared libraries like .a files, header files, images, and helper apps that support the library. Each layer has a set of frameworks that are helpful for developers.



Architecture of IOS

CORE OS Layer:

All the IOS technologies are built under the lowest level layer i.e. Core OS layer. These technologies include:

1. Core Bluetooth Framework
2. External Accessories Framework
3. Accelerate Framework
4. Security Services Framework
5. Local Authorization Framework etc.

It supports 64 bit which enables the application to run faster.

CORE SERVICES Layer:

Some important frameworks are present in the CORE SERVICES Layer which helps the iOS operating system to cure itself and provide better functionality. It is the 2nd lowest

layer in the Architecture as shown above. Below are some important frameworks present in this layer:

1. **Address Book Framework-**
The Address Book Framework provides access to the contact details of the user.
2. **Cloud Kit Framework-**
This framework provides a medium for moving data between your app and iCloud.
3. **Core Data Framework-**
This is the technology that is used for managing the data model of a Model View Controller app.
4. **Core Foundation Framework-**
This framework provides data management and service features for iOS applications.
5. **Core Location Framework-**
This framework helps to provide the location and heading information to the application.
6. **Core Motion Framework-**
All the motion-based data on the device is accessed with the help of the Core Motion Framework.
7. **Foundation Framework-**
Objective C covering too many of the features found in the Core Foundation framework.
8. **HealthKit Framework-**
This framework handles the health-related information of the user.
9. **HomeKit Framework-**
This framework is used for talking with and controlling connected devices with the user's home.
10. **Social Framework-**
It is simply an interface that will access users' social media accounts.
11. **StoreKit Framework-**
This framework supports for buying of contents and services from inside iOS apps.

MEDIA Layer:

With the help of the media layer, we will enable all graphics video, and audio technology of the system. This is the second layer in the architecture. The different frameworks of MEDIA layers are:

1. **UIKit Graphics-**
This framework provides support for designing images and animating the view content.
2. **Core Graphics Framework-**
This framework support 2D vector and image-based rendering and it is a native drawing engine for iOS.
3. **Core Animation-**
This framework helps in optimizing the animation experience of the apps in iOS.
4. **Media Player Framework-**
This framework provides support for playing the playlist and enables the user to use their iTunes library.
5. **AV Kit-**
This framework provides various easy-to-use interfaces for video presentation, recording, and playback of audio and video.

6. **Open AL-**

This framework is an Industry Standard Technology for providing Audio.

7. **Core Images-**

This framework provides advanced support for motionless images.

8. **GL Kit-**

This framework manages advanced 2D and 3D rendering by hardware-accelerated interfaces.

COCOA TOUCH:

Cocoa Touch is also known as the application layer which acts as an interface for the user to work with the iOS Operating system. It supports touch and motion events and many more features. The COCOA TOUCH layer provides the following frameworks :

1. **UIKit Framework-**

This framework shows a standard system interface using view controllers for viewing and changing events.

2. **GameKit Framework-**

This framework provides support for users to share their game-related data online using a Game Center.

3. **MapKit Framework-**

This framework gives a scrollable map that one can include in your user interface of the app.

4. **PushKit Framework-**

This framework provides registration support.

Features of iOS operating System:

Let us discuss some features of the iOS operating system-

1. Highly Securer than other operating systems.
2. iOS provides multitasking features like while working in one application we can switch to another application easily.
3. iOS's user interface includes multiple gestures like swipe, tap, pinch, Reverse pinch.
4. iBooks, iStore, iTunes, Game Center, and Email are user-friendly.
5. It provides Safari as a default Web Browser.
6. It has a powerful API and a Camera.
7. It has deep hardware and software integration

Applications of IOS Operating System:

Here are some applications of the iOS operating system-

1. iOS Operating System is the Commercial Operating system of Apple Inc. and is popular for its security.
2. iOS operating system comes with pre-installed apps which were developed by Apple like Mail, Map, TV, Music, Wallet, Health, and Many More.
3. Swift Programming language is used for Developing Apps that would run on IOS Operating System.
4. In iOS Operating System we can perform Multitask like Chatting along with Surfing on the Internet.

Advantages of IOS Operating System:

The iOS operating system has some advantages over other operating systems available in the market especially the Android operating system. Here are some of them-

1. More secure than other operating systems.
2. Excellent UI and fluid responsive
3. Suits best for Business and Professionals
4. Generate Less Heat as compared to Android.

Disadvantages of IOS Operating System:

Let us have a look at some disadvantages of the iOS operating system-

1. More Costly.
2. Less User Friendly as Compared to Android Operating System.
3. Not Flexible as it supports only IOS devices.
4. Battery Performance is poor.

SOFTWARE DEVELOPMENT TOOLS

1. iOS Software Development Kit (SDK):

- Released in 2008 for iPhone, iPad, and iPod touch development.
- Allows both experienced and novice developers to create iOS applications.
- Requires registration and a developer account at the iOS Dev Center.
- Only compatible with Intel-based Mac computers running Mac OS X Snow Leopard or later.

2. Objective C:

- Object-oriented programming language used for iOS app development.
- Adds object-oriented features to standard C with Smalltalk-style messaging.
- Emphasizes message passing over method calls.
- Supports reflection for querying and modifying program structure.
- Does not support operator overloading and multiple inheritance.

3. Xcode (Integrated Developer Environment):

- The primary IDE for iOS and Mac OS X app development.
- Includes developer tools and documentation.
- Uses a modified version of GCC to compile Objective C code.
- Supports debugging with GDB.
- Compiles iOS apps to run on ARM-based processors.
- Offers a feature called Dedicated Network Build for distributed compilation.

4. Interface Builder:

- Part of the Xcode toolset for creating app user interfaces.
- Provides a GUI-based interface for designing UI elements and connecting them to application code.

5. Instruments:

- A profiling and performance analysis tool for iOS apps.
- Uses the DTrace tracing framework.
- Tracks various performance metrics, including CPU load, memory allocation, graphics rendering, and user input events.

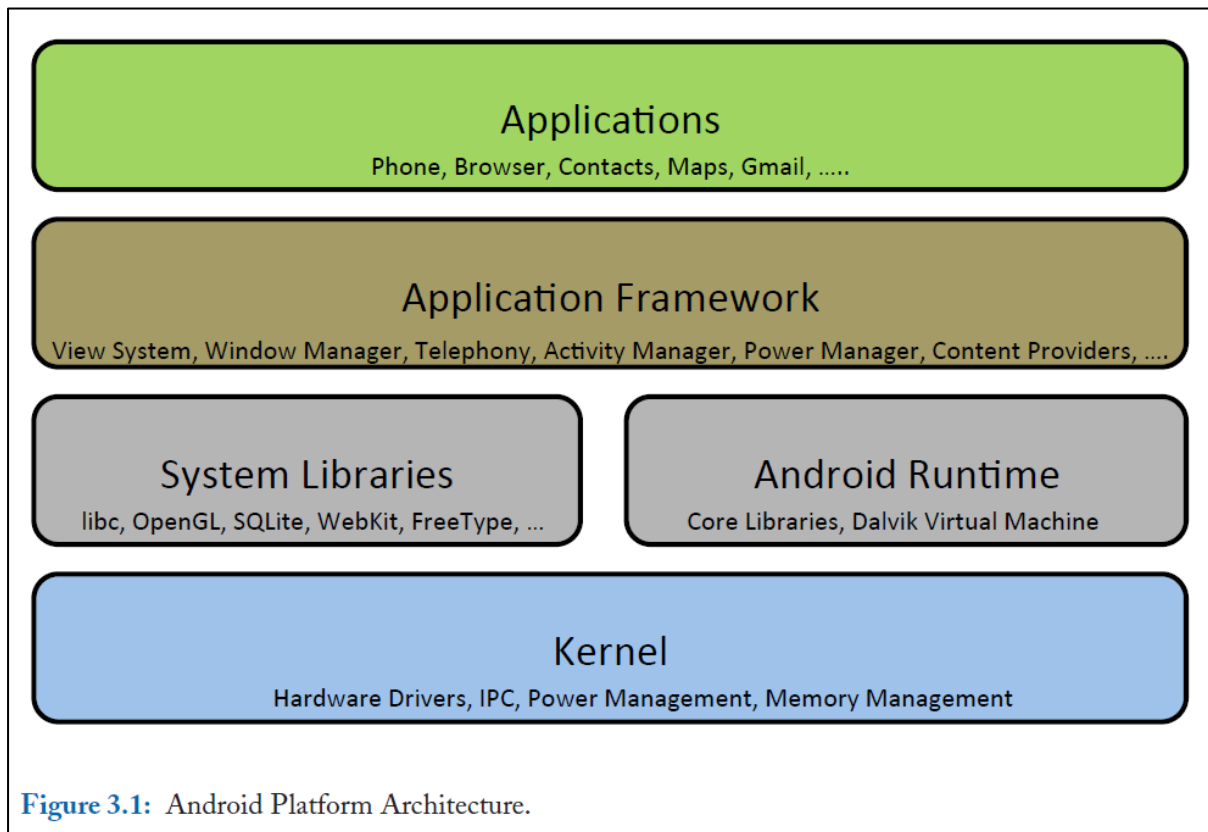
6. iOS Simulator:

- Allows running iOS apps on a Mac without deploying to a physical iOS device.
- Supports simulating iPhone and iPad devices with various iOS versions.
- Enables rapid testing, modification, and bug identification before actual deployment.
- Cannot simulate accelerometer sensor readings or camera images.

7. Writing Your First iOS App:

- Learning iOS development is encouraged through Xcode tutorials and sample app modification.
- The iOS simulator can be used for free to test different apps.
- Referring to example apps and source code is a helpful way to learn.

Android platform architecture



1. Kernel:

- Android uses a customized Linux kernel based on version 2.6.
- The kernel provides hardware abstraction, drivers, and lowlevel functionalities.
- It introduces WakeLocks to prevent devices from going into low power states, enhancing user experience.
- Android utilizes the Binder mechanism for interprocess communication and memory management.

2. Android Runtime:

- Android uses the Dalvik Virtual Machine (VM) for executing applications.
- Applications are written in a dialect of Java, compiled into .dex files, and run on the Dalvik VM.
- Each application runs in its own instance of the Dalvik VM for security and isolation.
- Android apps can request permissions to access device data during installation.

3. System Libraries:

- Android provides C/C++ libraries accessible through the Application Framework.
- Libraries include libc, FreeType, SQLite, OpenGL/ES, Scalable Graphics Library (SGL), and more.
- LibWebCore powers the Android browser engine, important for web applications.

- The Android Media Library handles multimedia content, including audio and video.

4. Application Framework:

- The Application Framework offers highlevel APIs for developers to use Android's capabilities.
- The View System is used to create user interfaces, with views, widgets, and view groups.
- ContentProvider enables data sharing between applications.
- NotificationManager manages notifications for events.
- Resources are managed separately for customization and efficiency.
- LocationManager handles locationbased services and GPS.
- InputMethodService allows developers to create custom input methods.
- TelephonyManager provides telephony service information.
- Utility classes simplify tasks like phone number formatting.
- PowerManager controls device power states and includes WakeLocks for responsiveness.

Android Anatomy

An Android application consists of four types of components:

1. Activity:

- Represents a screen with a visual user interface.
- Defined by the Activity class.
- An app can have multiple activity components for different screens.
- Activities can be started in other applications asynchronously through Intents.

2. Service:

- Used for background tasks and functionalities that don't require direct user interaction.
- Defined by the Service class.
- Does not have a user interface.
- Can be started based on requests from other applications.

3. Content Provider:

- Enables data sharing between applications.
- Defined by the ContentProvider class.
- Accessed using the ContentResolver interface.
- Other apps can query and, if allowed, modify data provided by a content provider.
- Interaction with a content provider is not initiated through Intent objects.

4. Broadcast Receiver:

- Responds to system-wide broadcast announcements.
- Defined by the BroadcastReceiver class.
- Handles system status messages and events.
- Represents an Intent object that carries the broadcast message.
- Can create notifications on the device's user interface using the Notification Manager.

All these components must be declared in the AndroidManifest.xml file, including any declared Intent Filters to specify their capabilities. The manifest file also defines required user permissions, hardware and software services, external libraries, and the API Level used by the application. It plays a crucial role in specifying the application's configuration and capabilities.