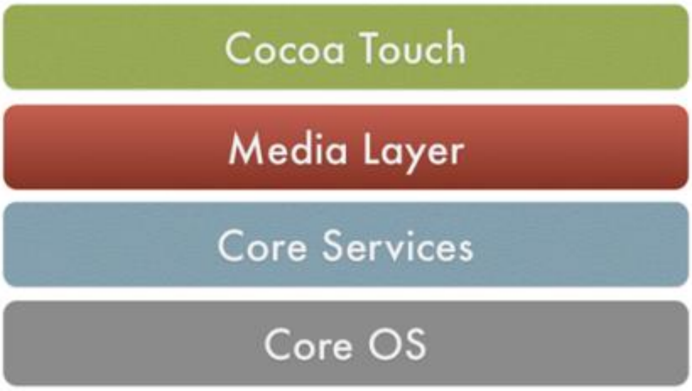
Architecture of iOS

Architecture of IOS is a layered architecture. At the uppermost level iOS works as an intermediary between the underlying hardware and the apps you make. Apps do not communicate to the underlying hardware directly.
Apps talk with the hardware through a collection of well defined system interfaces. These interfaces make it simple to write apps that work constantly on devices having various hardware abilities.
Lower layers gives the basic services which all application relies on and higher level layer gives sophisticated graphics and interface related services.
Apple provides most of its system interfaces in special packages called frameworks. A framework is a directory that holds a dynamic shared library that is .a files, related resources like as header files, images, and helper apps required to support that library. Every layer have a set of Framework which the developer use to construct the applications.



1. Core OS Layer:
The Core OS layer holds the low-level features that most other technologies are built upon.

- Core Bluetooth Framework.
- Accelerate Framework.
- External Accessory Framework.
- Security Services framework.

- Local Authentication framework.

64-Bit support from IOS7 supports the 64-bit app development and enables the application to run faster.

2. Core Services Layer

Some of the Important Frameworks available in the core services layers are detailed:

- **Address book framework** – Gives programmatic access to a contacts database of user.
- **Cloud Kit framework** – Gives a 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 give fundamental data management and service features for iOS apps.
- **Core Location framework –** Gives location and heading information to apps.
- **Core Motion Framework –** Access all motion-based data available on a device. Using this core motion framework Accelerometer based information can be accessed.
- **Foundation Framework –** Objective C covers too many of the features found in the Core Foundation framework
- **Healthkit framework –** New framework for handling health-related information of the user
- **Homekit framework –** A new framework for talking 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 is known as In-App Purchase.

3. Media Layer: Graphics, Audio, and Video technology is enabled using the Media Layer.

Graphics Framework:

- **UIKit Graphics –** It describes high-level support for designing images and is also used for animating the content of your views.
- **Core Graphics framework –** It is the native drawing engine for iOS apps and gives support for custom 2D vector and image-based rendering.
- **Core Animation –** It is an initial technology that optimizes the animation experience of your apps.
- **Core Images –** gives advanced support for controlling video and motionless images in a non-destructive way
- **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. It offers very low overhead access to the A7 GPU.

Read these iOS Interview Questions to grab high-paying jobs!

Audio Framework:

- **Media Player Framework –** It is a high-level framework that gives simple use to a user’s iTunes library and support for playing playlists.
- **AV Foundation –** It is an Objective C interface for handling the recording and playback of audio and video.
- **OpenAL –** is an industry-standard technology for providing audio.

Video Framework

- **AV Kit –** framework gives a collection of easy-to-use interfaces for presenting video.
- **AV Foundation –** gives advanced video playback and recording capability.
- **Core Media –** framework describes the low-level interfaces and data types for operating media.

Cocoa Touch Layer

- **EventKit framework –** gives view controllers for showing the standard system interfaces for seeing and altering calendar-related events
- **GameKit Framework –** implements support for Game Center which allows users share their game-related information online
- **iAd Framework –** allows you to deliver banner-based advertisements from your app.
- **MapKit Framework –** gives a scrollable map that you can include in your user interface of the app.
- **PushKitFramework –** provides registration support for VoIP apps.
- **Twitter Framework –** supports a UI for generating tweets and support for creating URLs to access the Twitter service.
- **UIKit Framework –** gives a vital infrastructure for applying graphical, event-driven apps in iOS. Some of the Important functions of UI Kit framework:

-Multitasking support.
– Basic app management and infrastructure.
– User interface management
– Support for Touch and Motion event.
– Cut, copy and paste support, and many more.