**Core Service Layer**

* Core Service Layer is Fundamental System Service for apps.
* The Service key are Core foundation and Foundation Frameworks.
* This Layer also Contains technology to support feature such as location, iCloud, social media, and networking.

**Features Of Core OS Layers:**

**i) Peer -to Peer Services:**

* It relating to networks in which each computer can act as a server for the others.
* The Multipeer Connectivity Framework provide peer to peer Conectivity Over . The Bluetooth standard provides a peer-to-peer (P2P) data exchange functionality over short distance between devices.
* Peer to Peer is used in games.

**ii)icloud Storage**:

* icloud storage of your app write user document and data in central location.User can access those items from all computer and devices.
* Using icloud means user can view or edit those document from any devices without doing file transfer explicitly.
* Storing Document in user icloud account it is saftey for user .If lose device documents are not lost even if they are in the icloud storage.

iii)  **Block Objects:**

* Block objects are a C-level syntactic and runtime feature that allow you to compose function expressions that can be passed as arguments, optionally stored, and used by multiple threads.
* Blocks are particularly useful as callbacks or in places where you need a way of easily combining both the code to be executed and the associated data.

**iOS Blocks are used:**

* As a replacement for delegates and delegate methods
* As a replacement for callback functions

**iv) Data Protection:**

* Data protection is available for devices that offer hardware encryption, including iPhone 3GS and later, all iPad models, and iPod touch (3rd generation and later).
* Data protection enhances the built-in hardware encryption by protecting the hardware encryption keys with your passcode. This provides an additional layer of protection for your email messages attachments, and third-party applications
* Implementing data protection it requires how you create and manage data you want to protect

**v) File SharingSupport:**

* An App support file sharing makes the contents the documents directory available to user.
* The user can move the file in and Out whenever needed from itunes.

**The file sharing for app are:**

* Put whatever files you want to share in your app’s Documents directory.
* When the device is plugged into the user’s computer, iTunes displays a File Sharing section in the Apps tab of the selected device.
* The user can add files to this directory or move files to the desktop.
* Apps that support file sharing should be able to recognize when files have been added to the Documents directory and respond.

**vi)Grand central Dispatch:**

* It is technology developed by Apple.
* Application support for systems with [multi-core](https://en.wikipedia.org/wiki/Multi-core_(computing)) processors and other [symmetric multiprocessing](https://en.wikipedia.org/wiki/Symmetric_multiprocessing) systems.
* GCD was first released with [Mac OS X 10.6](https://en.wikipedia.org/wiki/Mac_OS_X_10.6), and is also available with [iOS](https://en.wikipedia.org/wiki/IOS_(Apple)) 4.
* It is also provide reading, writing, file descriptors, implementing timers and monitoring Signal and process events.

**vii) In App purchase:**

* The purchase of goods and services from an application on a mobile device such as a Smartphone or tablet.
* In-app purchases allow developers to provide their application for free, while providing anyone who downloads the free version the opportunity to upgrade.
* Because the developer can always add new features and content to the application after it is downloaded, application users won’t have to download a different application.
* These feature implemented from StoreKit framework.

**viii)SQLite**

* SQLite can be used in iOS for handling data. It uses sqlite queries, which makes it easier for those who know SQL.
* From An app create an local database files and managed tables and record in those files.
* The header file for accessing the SQLite library is located in **<iOS\_SDK>/usr/include/sqlite3.h**
* where <iOS\_SDK> is the path to the target SDK in your Xcode installation directory.

**ix) XML Support:**

* The Foundation framework provides the [NSXMLParser](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSXMLParser_Class/index.html#//apple_ref/occ/cl/NSXMLParser) class for retrieving elements from an XML document.
* Additional support we get from XML by libxml2 library.
* This Open source library you write XML data quickly and transform XML content to HTML.
* The header files for accessing the libxml2 library are located in <iOS\_SDK>/usr/include/libxml2/ directory,
* where <iOS\_SDK> is the path to the target SDK in your Xcode installation .

**Core Service Framework:**

**i)Account Framework:**

* It is used to for single sign for certain user accounts.
* The Account framework provide used access user account stored in Account database.

**ii)Address BookFramework:**

* The Address framework used to provide programmatic access to user contact database.
* If your app uses Contact Information By using this framework you can modify that information.

**iii)Ad Support Framework:**

* It Provides apps with access to an identifier that can be used for advertsing purposes.
* This is used for advertising purposes.
* The Ad Support framework provides access to an identifier that apps can use for advertising purposes.
* This framework also provides a flag that indicates whether the user has option out of ad tracking.
* Apps are required to read and honor the opt-out flag before trying to access the advertising identifier.

### iv) CFNetwork Framework

* This can be used for working with network protocols.
* The CFNetwork framework is a set of high-performance C-based interfaces that use object-oriented abstractions for working with network protocols.
* These abstractions give you detailed control over the protocol stack and make it easy to use lower-level constructs such as BSD sockets.
* You can use this framework to simplify tasks such as communicating with FTP and HTTP servers or resolving DNS hosts. With the CFNetwork framework, you can:
* Use BSD sockets
* Create encrypted connections using SSL or TLS
* Resolve DNS hosts
* Work with HTTP servers, authenticating HTTP servers, and HTTPS servers
* Work with FTP servers
* Publish, resolve, and browse Bonjour services
* CFNetwork is based, both physically and theoretically, on BSD sockets.

### v)CloudKit Framework

* CloudKit provides a conduit for moving data between the app and iCloud.
* Unlike other iCloud technologies where data transfers happen transparently, CloudKit gives you control over when transfers occur.
* You can use CloudKit to manage all types of data.
* Apps that use CloudKit directly can store data in a repository that is shared by all users.
* This public repository is tied to the app itself and is available even on devices without a registered iCloud account.
* As the app developer, we can manage the data in this container directly and see any changes made by users through the CloudKit dashboard.

### vi)Core Data Framework

* The Core Data significantly reduces the amount of code that we have to write for the app.
* The Core Data framework is a technology for managing the data model of a Model-View-Controller app.
* Core Data is intended for use in apps in which the data model is already highly structured.
* Instead of defining data structures programmatically, we can use the graphical tools in Xcode to build a schema representing your data model.
* At runtime, instances of the data-model entities are created, managed, and made available through the Core Data framework.

**The Core Data also provides the following features:**

* Storage of object data in a SQLite database for optimal performance.
* An [NSFetchedResultsController](https://developer.apple.com/library/ios/documentation/CoreData/Reference/NSFetchedResultsController_Class/index.html#//apple_ref/occ/cl/NSFetchedResultsController) class to manage results for table views.
* Management of undo/redo beyond basic text editing.
* Support for the validation of property values.
* Support for propagating changes and ensuring that the relationships between objects remain consistent.
* Support for grouping, filtering, and organizing data in memory.

### vii) Core Foundation Framework

* The Core Foundation framework is a set of C-based interfaces that provide basic data management and service features for iOS apps.

This framework includes support for the following:

* Collection data types (arrays, sets, and so on)
* Bundles
* String management
* Date and time management
* Raw data block management
* Preferences management
* URL and stream manipulation
* Threads and run loops
* Port and socket communication
* The Core Foundation framework is closely related to the Foundation framework, which provides Objective-C interfaces for the same basic features.
* When we need to mix Foundation objects and Core Foundation types, we can take advantage of the “toll-free bridging” that exists between the two frameworks.
* Toll-free bridging means that we can use some Core Foundation and Foundation types interchangeably in the methods and functions of either framework.
* This support is available for many of the data types, including the collection and string data types.

### viii)Core Location Framework

* The Core Location framework provides location and heading information to the apps.
* For location information, the framework uses the onboard GPS, cell, or Wi-Fi radios to find the user’s current longitude and latitude.
* We can incorporate this technology into our own apps to provide position-based information to the user.
* For example, we might have a service that searches for nearby restaurants, shops, or facilities, and base that search on the user’s current location.

**Core Location also provides the following capabilities:**

* Access to compass-based heading information on iOS devices that include a magnetometer
* Support for region monitoring based on a geographic location or Bluetooth beacon
* Support for low-power location-monitoring using cell towers
* Collaboration with MapKit to improve the quality of location data in specific situations, such as when driving

### ix) Core Media Framework

* The Core Media framework provides the low-level media types used by the AV Foundation framework.
* Most apps never need to use this framework, but it is still provided for those few developers who need more accurate control over the creation and presentation of audio and video content.

### x)Core Motion Framework

* The Core Motion framework provides a single set of interfaces for accessing all motion-based data available on a device.
* This framework supports accessing both raw and processed accelerometer data using a new set of block-based interfaces.
* For devices with a built-in gyroscope, you can retrieve the raw gyro data as well as processed data reflecting the attitude and rotation rates of the device.
* We can use both the accelerometer and the gyro-based data for games or other apps that use motion as input or as a way to enhance the overall user experience.
* For devices with step-counting hardware, we can access that data and use it to track fitness-related activities.

### xi)Core Telephony Framework

* The Core Telephony framework provides interfaces for interacting with phone-based information on devices that have a cellular radio.
* Apps can use this framework to get information about a user’s cellular service provider.
* Apps interested in cellular call events (such as VoIP apps) can also be notified when those events occur.

### xii)EventKit Framework

* The EventKit framework provides an interface for accessing calendar events on a user’s device.

**We can use this framework to do the following:**

* Get existing events and reminders from the user’s calendar
* Add events to the user’s calendar
* Create reminders for the user and have them appear in the Reminders app
* Configure alarms for calendar events, including setting rules for when those alarms should be triggered

### xiii)Foundation Framework

* The Foundation framework provides Objective-C wrappers to many of the features found in the Core Foundation framework, which is described in [Core Foundation Framework](https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html#//apple_ref/doc/uid/TP40007898-CH10-SW1).

**Foundation framework provides support for the following features:**

* Collection data types (arrays, sets, and so on)
* Bundles
* String management
* Date and time management
* Raw data block management
* Preferences management
* URL and stream manipulation
* Threads and run loops
* Bonjour
* Communication port management
* Internationalization
* Regular expression matching
* Cache support

### xiv)HealthKit Framework

* The HealthKit framework is a new framework for managing a user’s health-related information.
* With the proliferation of apps and devices for tracking health and fitness information, it's difficult for users to get a clear picture of how they are doing.
* HealthKit makes it easy for apps to share health-related information, whether that information comes from devices connected to an iOS device or is entered manually by the user.
* The user’s health information is stored in a centralized and secure location.
* The user can then see all of that data displayed in the Health app.
* When the app implements support for HealthKit, it gets access to health-related information for the user and can provide information about the user, without needing to implement support for specific fitness-tracking devices.
* The user can decide which data should be shared with his app.
* Once data is shared with the app, the app can register to be notified when that data changes; we have fine-grained control over when your app is notified.
* For example, you could request that your app be notified whenever the user takes his or her blood pressure, or be notified only when a measurement shows that the user’s blood pressure reaches a specific reading.

### xv) HomeKit Framework

* HomeKit is a new framework for controlling the devices in home.
* New devices are being introducing for the home to offering more connectivity between mobile and devices.
* HomeKit framework provides a standardized way to communicate with those devices.
* By using this framework in App user can communicate with devices that users have in their homes.
* Using this kind of app, users can discover devices in their home and configure them.
* User can also create actions to control those devices.
* The user can group actions together and trigger them using the App called Siri.
* Once a configuration is created, users can invite other people to share access to it. For example, a user might temporarily offer access to a house guest.
* Use the HomeKit Accessory Simulator to test the communication of your HomeKit app with a device.

### xvi)JavaScript Core Framework

* The JavaScript Core framework provides Objective-C wrapper classes for many standard JavaScript objects.
* Use this framework to evaluate JavaScript code and to parse JSON data.

### xvii)Mobile Core Services Framework

* The Mobile Core Services framework defines the low-level types used in uniform type identifiers (UTIs).
* UTI provides a unique identifier for a particular file type, data type, directory or bundle type, and so on.

### xviii)Multipeer Connectivity Framework

* The Multipeer Connectivity framework supports to find of nearby devices and the direct communication with those devices without requiring Internet connectivity.
* By using this framework we can create Multipeer sessions easily and to support reliable in-order data transmission and real-time data transmission.
* Because of this framework, the app can communicate with nearby devices and can exchange unlimited data.
* This framework provides programmatic and UI-based options for discovering and managing network services.
* This App can integrate the [MCBrowserViewController](https://developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MCBrowserViewController_class/index.html#//apple_ref/occ/cl/MCBrowserViewController) class into their UI to display a list of peer devices for the user to choose from.
* Alternatively, we can use the [MCNearbyServiceBrowser](https://developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MCNearbyServiceBrowserClassRef/index.html#//apple_ref/occ/cl/MCNearbyServiceBrowser) class to look for and manage peer devices programmatically.

### xix)NewsstandKit Framework

* The Newsstand app is available for iPhone provides a central place for users to read magazines and newspapers.
* Publishers who want to deliver their magazine and newspaper content through Newsstand can create their own iOS apps using the NewsstandKit framework which lets you initiate background downloads of new magazine and newspaper issues.
* After you start a download, the system handles the download operation and notifies your app when the new content is available.

### xx)PassKit Framework

* The Passbook app provides a place to store coupons for users like boarding passes, event tickets, and discount cards for businesses.
* Instead of carrying a physical representation of the above mentioned items, users can now store them into their iOS device and use them the same way as before they are using.
* The PassKit framework provides the Objective-C interfaces you need to integrate support for these items into your apps
* We can use this framework in combination with web interfaces and file format information to create and manage the passes that company offers.
* Passes are created by the company’s web service and delivered to their user’s device via email, Safari, or their custom app.
* The pass itself, using a special file format like cryptographically signed before being delivered to the user.
* The file format identifies relevant information about the service being offered so that the user knows what the service is for.
* It might also contain a bar code or other information that we can then use to validate the card so this can be redeemed or used by many companies.

### xxi)Quick Look Framework

* The Quick Look framework is used for previewing the contents of App directly.
* The main primary reason of this framework is for apps that, download files from the network or that otherwise work with files from unknown sources.
* After obtaining the file, we can use the view controller provided by this framework to display the contents of that file directly in the user interface.

### xxii)Safari Services Framework

* This Safari Services framework provides support for programmatically adding URLs to the user’s Safari reading list.

### xxiii)Social Framework

* This Social framework provides a simple interface for accessing the user’s social media accounts like Facebook and Twitter.
* This framework covers the Twitter framework and adds support for other social accounts, including Facebook, Sina Weibo, and others.
* All Apps can use this framework to post status updates and images to the user’s account.
* This framework works with the Accounts framework to provide a single sign-on model for the user and to ensure that access to the user’s account is approved.

### xxiv)StoreKit Framework

* The StoreKit framework provides support for the purchasing of content and services from within our iOS apps this feature is known as In-App Purchase
* For example, by using this feature we can give a feature to allow the user to unlock additional app features that are available in App.
* If we are developing a game, we can use it to offer additional game levels.
* In both cases, the StoreKit framework handles the financial aspects of the transaction and processing payment requests through the user’s iTunes Store account and providing the app with information about the purchase.
* The StoreKit framework focuses on the financial aspects of a transaction, ensuring that transactions occur securely and correctly.
* Our app handles the other aspects of the transaction, including the presentation of a purchasing interface and downloading or unlocking of the appropriate content.
* This division of labor gives you control over the user experience for purchasing content.
* When we decided to go through these kinds of applications like purchasing interface you for the user we can make use of this framework.
* We can also decide the delivery mechanism that works best for our app.

### xxv)System Configuration Framework

* The System Configuration framework is used to provide the reach ability interfaces like we can use to determine the network configuration of a device.
* We can use this framework to determine whether a Wi-Fi or cellular connection is in use and whether a particular host server can be accessed or not.

### xxvi)WebKit Framework

* The WebKit framework is used to display the HTML contents in App.
* In addition to displaying HTML, we can provide basic editing support so that users can replace text and manipulate document text and attributes, including CSS properties.
* WebKit also supports creating and editing content at the DOM level of an HTML document.
* For example, you could extract the list of links on a page, modify them, and replace them prior to displaying the document in a web view.