**Core OS Layer**

* The Core OS Layer it contains low level features that other technology are built.
* You not use in your app means then also used by other frameworks.
* In some situation you need to deal with security or communicating with an external hardware accessors you have to use frameworks in this layer.
* **Accelerate Framework :**
* The Accelerate Framework contains interface for performing digital signal processing, linear algebra, and image processing calculations.
* The main advantage of using this framework by writing own version of interfaces that they are optimized for all hardware configuration in iOS device.
* Therefore we code once and run efficiently in all devices.
* **Core Bluetooth framework:**
* It is used to interact with Bluetooth low energy accessories.(LE) i.e( is a power-conserving variant of [Bluetooth](http://searchmobilecomputing.techtarget.com/definition/Bluetooth) personal area network ([PAN](http://searchmobilecomputing.techtarget.com/definition/personal-area-network)) technology, designed for use by Internet-connected machines and appliances).

**Objective C interfaces used for the following framework:**

* Scan for Bluetooth accessories and Connect and Disconnect after it displayed.
* Vend from app ,one iOS device into other Bluetooth devices.
* when app is launched preserve state of Bluetooth connection and restore connection.
* Give Notification for changes availability of Bluetooth peripheral.
* **External Accessory Framework:**
* It provide support for Communicating with hardware accessories attached to an iOS device.
* Accessories can be connected through the 30-pin dock connector of a device or wirelessly using Bluetooth.
* External Accessory framework get information abt accessory and initiate communication session
* **General Security Services Frameworks:**
* It provides a Standard set of security related services to iOS apps.
* The basic interfaces of this framework are specified in IETF [RFC 2743](http://www.ietf.org/rfc/rfc2743.txt) and [RFC 4401](http://tools.ietf.org/html/rfc4401).
* **Local Authentication Framework:**
* Local Authentication framework use touch ID to authenticate the user.
* Some of app required secure access to all of their content. some of apps its not required its optional.
* But in both cases user to authenticate before proceeding.
* Use this framework to display an alert to the user with an application.
* **Network Extension Framework:**
* The Network Extension framework support for configuring and controlling Virtual private network.
* Use this framework to create VPN Configuration
* **Security Framework:**
* ios gives security framework for managing security of your apps.
* This framework provide interfaces for managing certificates, public and private keys.
* **System:**
* The system level encompasses the kernel environment, drivers, and low-level UNIX interfaces of the operating system.
* The kernel itself, based on Mach, is responsible for every aspect of the operating system. It manages the virtual memory system, threads, file system, network, and interprocess communication.
* The drivers at this layer also provide the interface between the available hardware and system frameworks.
* iOS provide interfaces for low level feature of operating system.
* App can access these features through Libsystem library.

C based interfaces are giving support given below:

* Concurrency (POSIX threads and Grand Central Dispatch)
* Networking (BSD sockets)
* File-system access
* Standard I/O
* Bonjour and DNS services
* Locale information
* Memory allocation
* Math computations

Header files for many Core OS technologies are located in the <iOS\_SDK>/usr/include/ directory, where <iOS\_SDK> is the path to the target SDK in your Xcode installation directory.

# 64 Bit Support:

* iOS was initially designed to support binary files on devices using a 32 bit architecture.
* The support was introduced for compling, linking, an debugging binaries on 64 bit architecture.
* system libraries and frameworks that they can be used in both 32-bit and 64-bit apps.
* when compiled for 64-bit runtime the apps may run faster because its having extra processor in 64 bit mode.
* iOS uses the LP64 model that is used by OS X and other 64-bit UNIX systems.