

IS|S-3510

Mobile App Development **3.0**



(a.k.a., *Software Engineering for Mobile Applications*)

THE iOS ARCHITECTURE



iOS

- Proprietary Operating System.
- Designed/Created by Apple to support most of their mobile devices (iPads now use iPadOS).
- Is different from MacOS because of Cocoa Touch (the nearest layer to the user).
- Newest version: iOS 14 (2020)

**WHICH IS THE IOS
KERNEL?**

iOS

IOS is CISCO's routers operating system, do not mistake it with iOS

- iOS is a UNIX relative
- iOS kernel is based on Mach
- Mach is Darwin's kernel (BSD variant)

[iPhone OS 1.0 - iPhone OS 3.13](#)

[iOS4](#)

[iOS 5](#)

[iOS 6](#)

[iOS 7, iOS 8](#)

[iOS 9](#)

[iOS 10](#)

Darwin 9.0.0d1

Darwin 10

Darwin 11

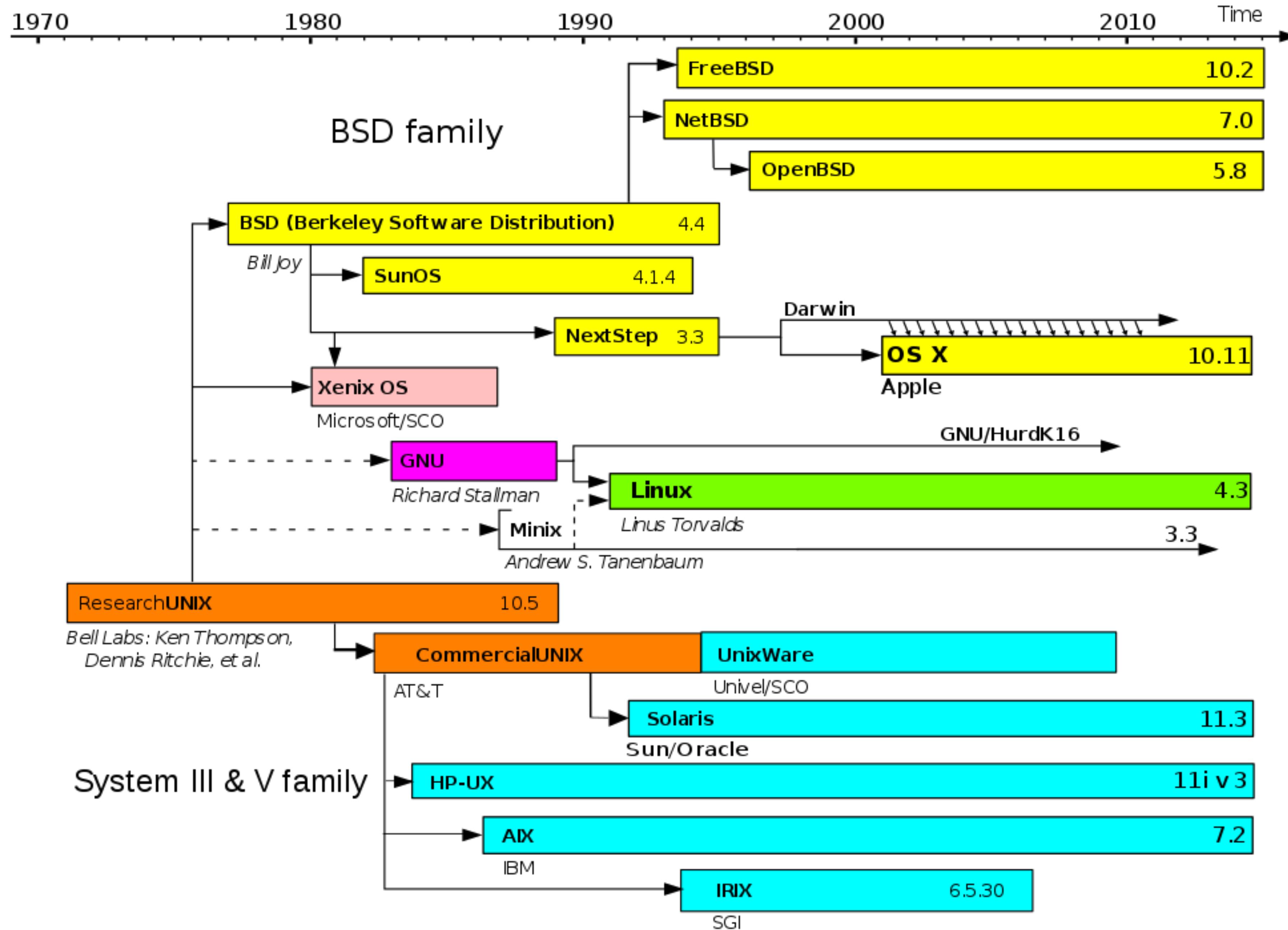
Darwin 13

Darwin 14

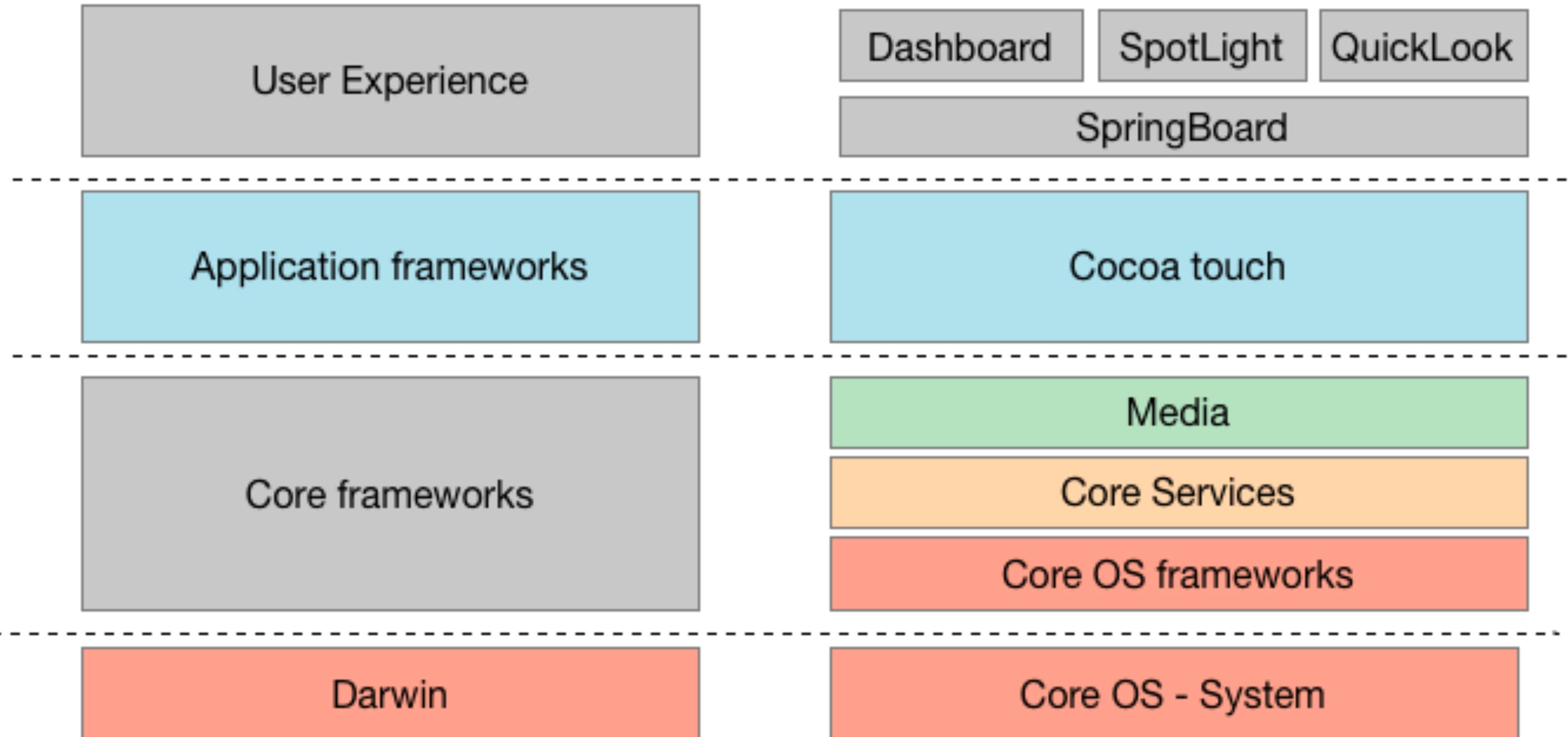
Darwin 15

Darwin 16

UNIX TIMELINE



THE iOS STACK



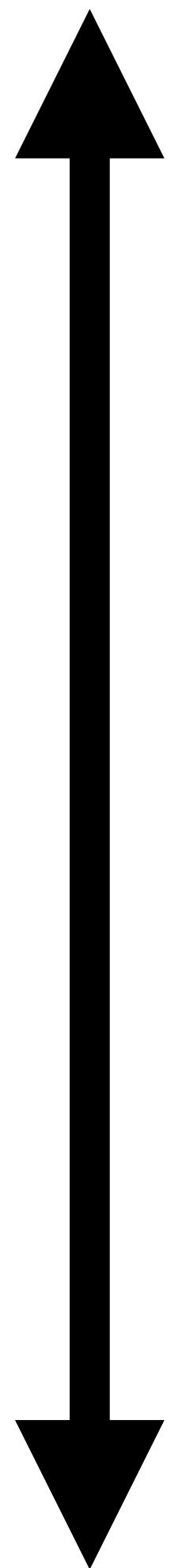
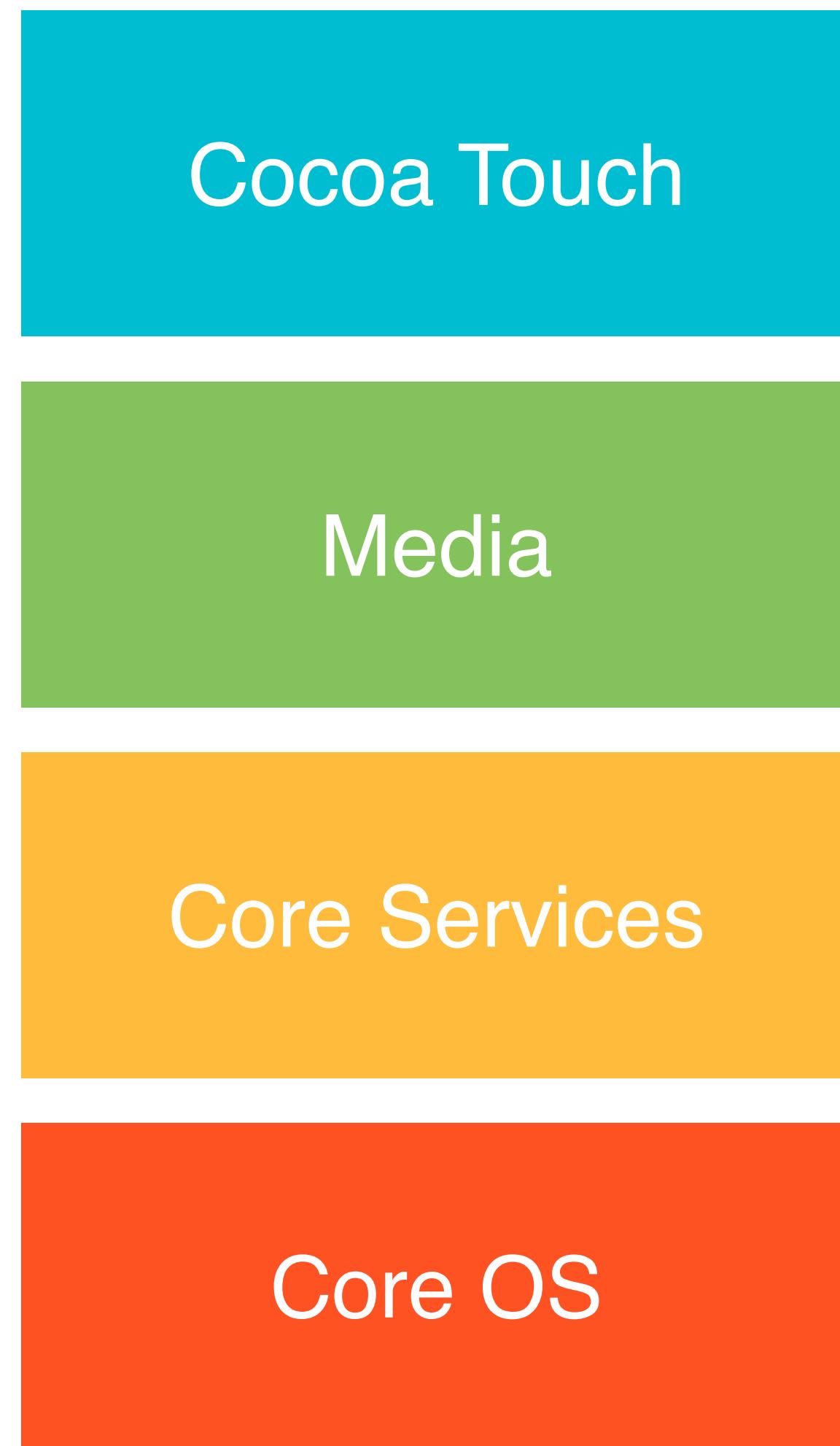


The top layer should interact with the user, e.g., apps layer in Android, or user experience layer in iOS

A second layer should provide high-level frameworks/libraries/services the apps rely on

WHAT IS THE
DIFFERENCE BETWEEN
USER AND KERNEL
SPACE?

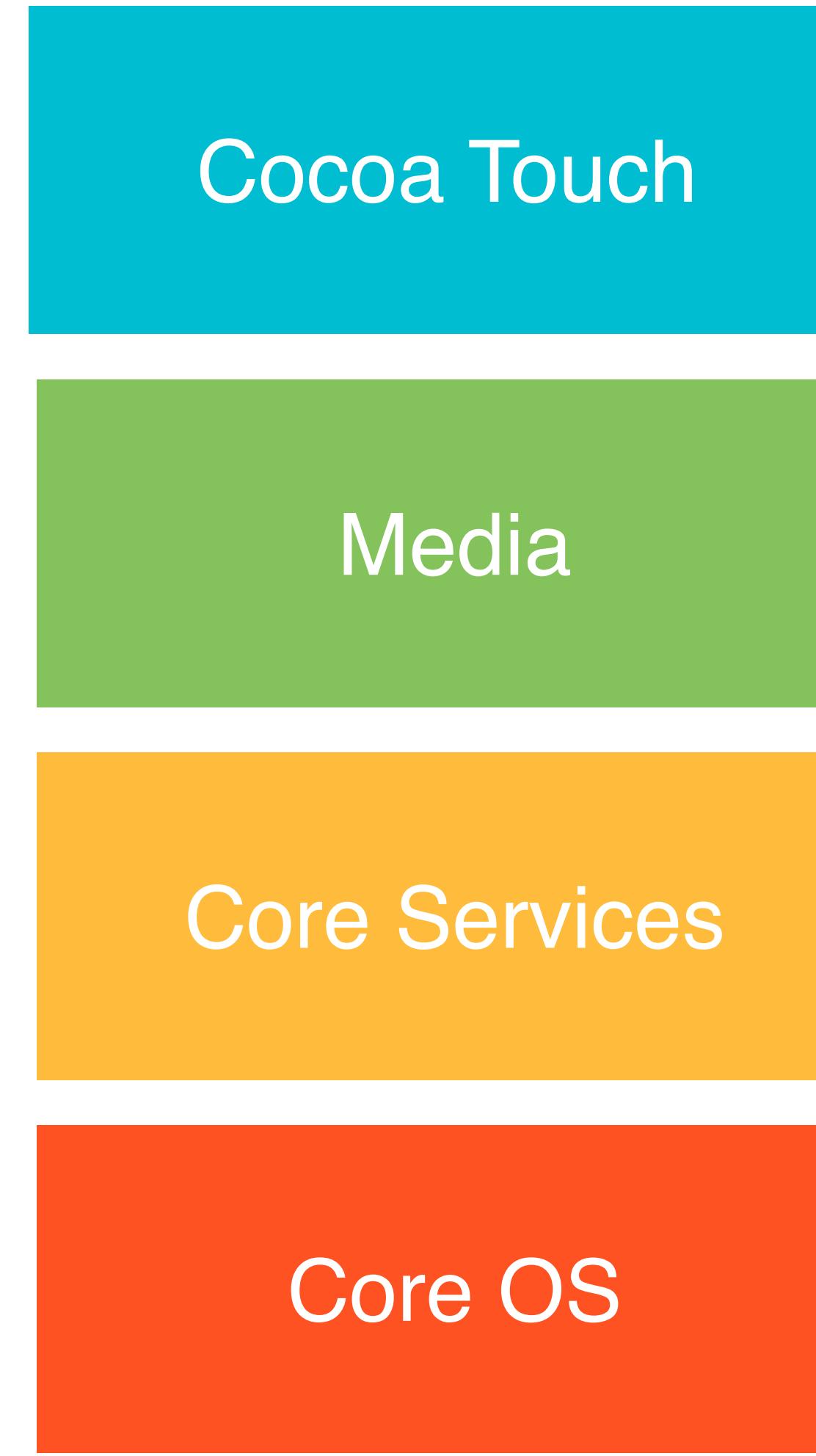
THE iOS STACK



USER

HARDWARE

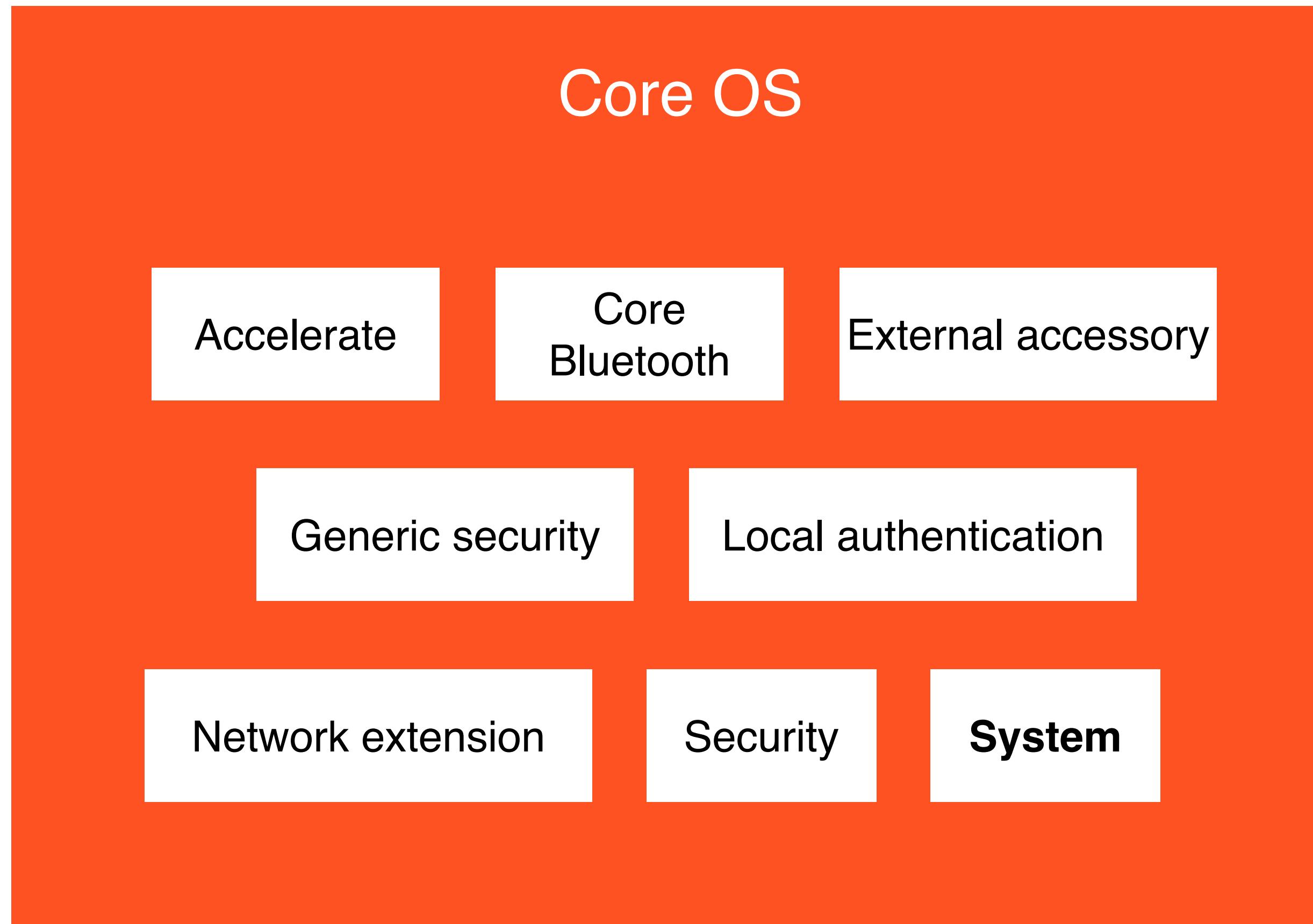
THE iOS STACK



- Top layers make use of the bottom layers
- Top layers are more abstract and require less low level programming
- Developers have access to all the layers.
- Each layer has a set of frameworks that display “high-level features”
- Apps/layers use interfaces of each layer

**WHERE IS THE KERNEL
OF THE OS?**

THE iOS STACK: CORE OS

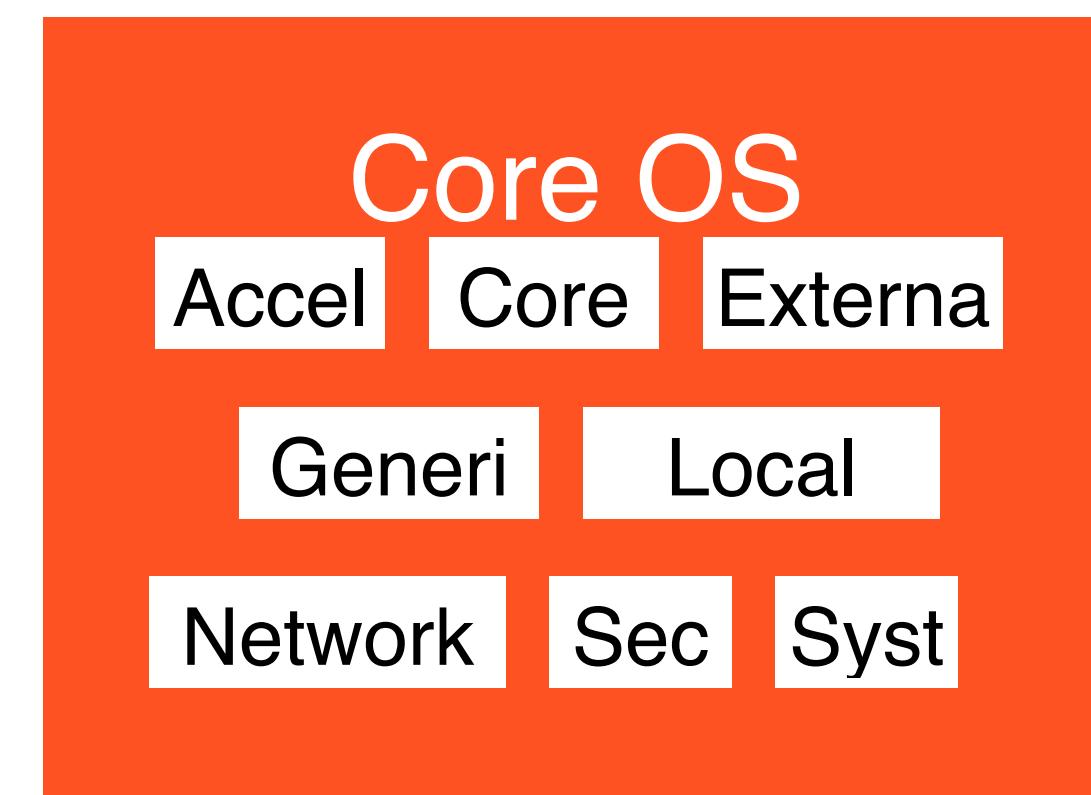


It has low level functionalities/services required by the other layers

Networking, memory management, file system, threading, sensors, etc.

System: kernel, drivers, low level UNIX interface.

THE iOS STACK: CORE OS

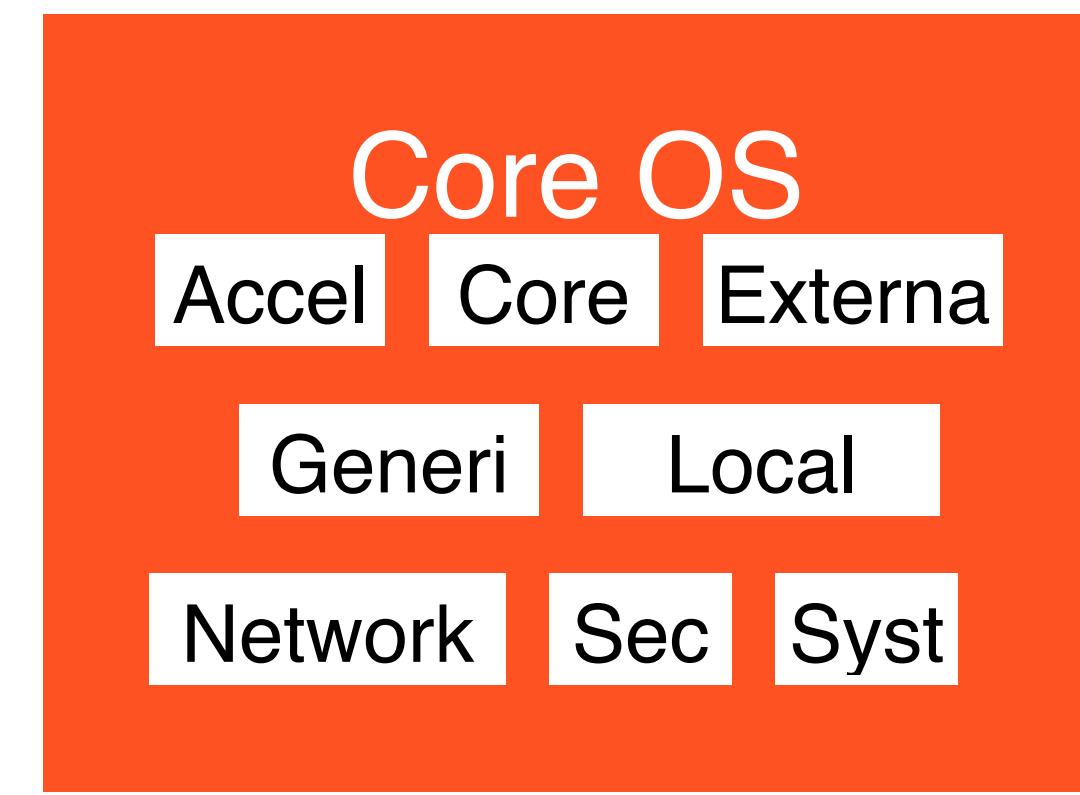


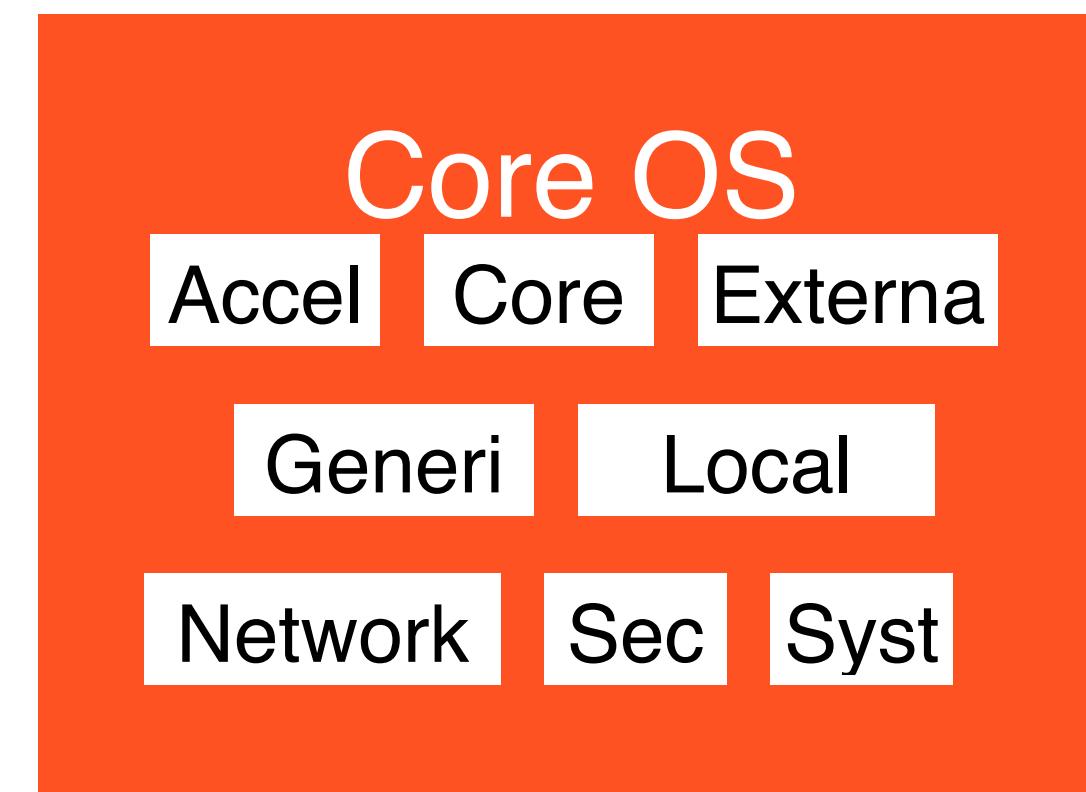
Accelerate (Accelerate.framework):

- API based on C and optimized for the device hardware.
- Image processing
- Signals processing
- Vectors (Linear Algebra)
- Mathematics for complexes and large numbers

Core bluetooth (CoreBluetooth.framework):

- Interfaces written on Objective-C
- Interactions with Bluetooth LE accessories





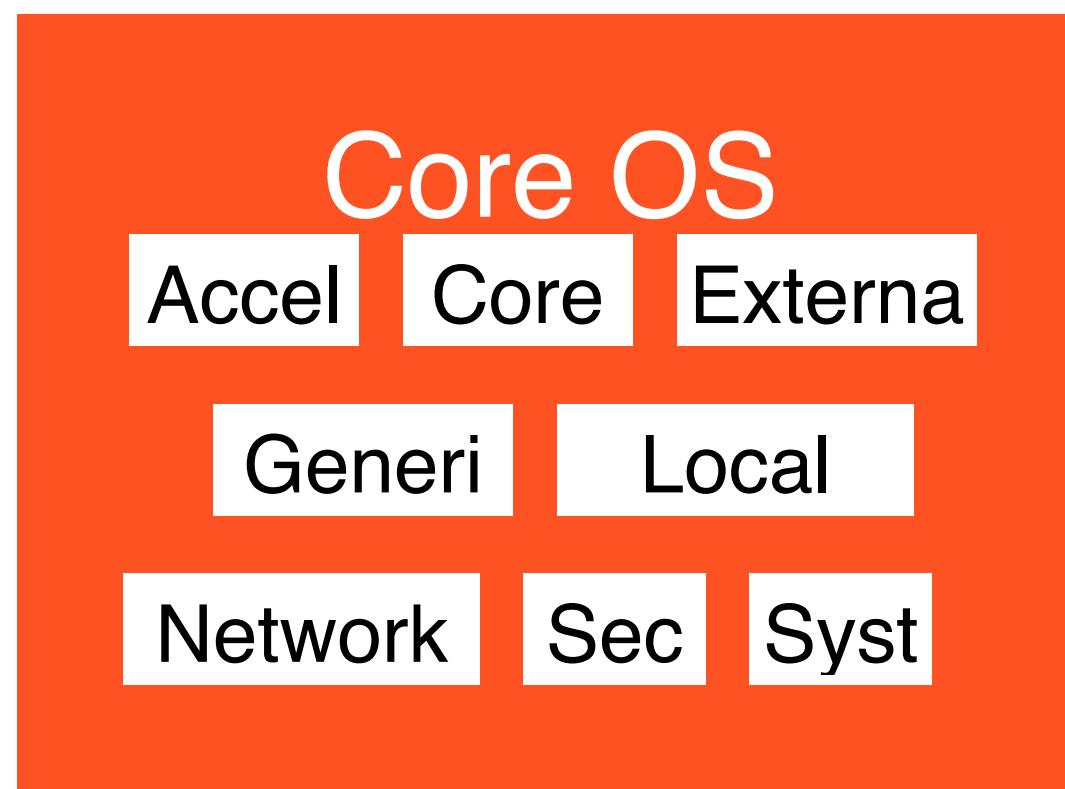
External Accessory (ExternalAccessory.framework):

- Communication with external devices connected by cable or bluetooth

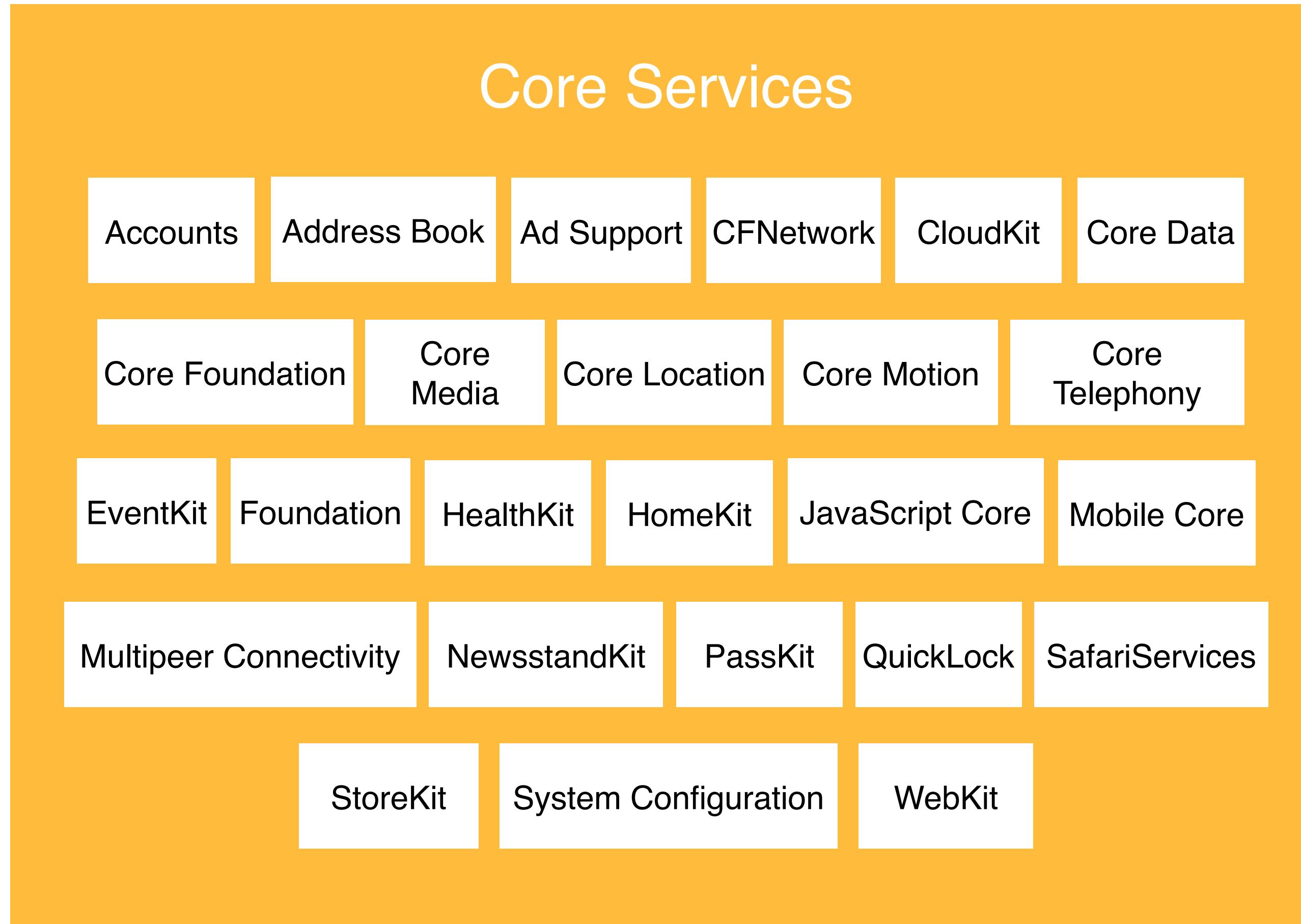
THE iOS STACK: CORE OS

Interfaces based on C (LibSystem):

- Concurrency
- Networking
- Access to the file system
- I/O standard
- Bonjour DNS
- Memory management
- Mathematical calculations

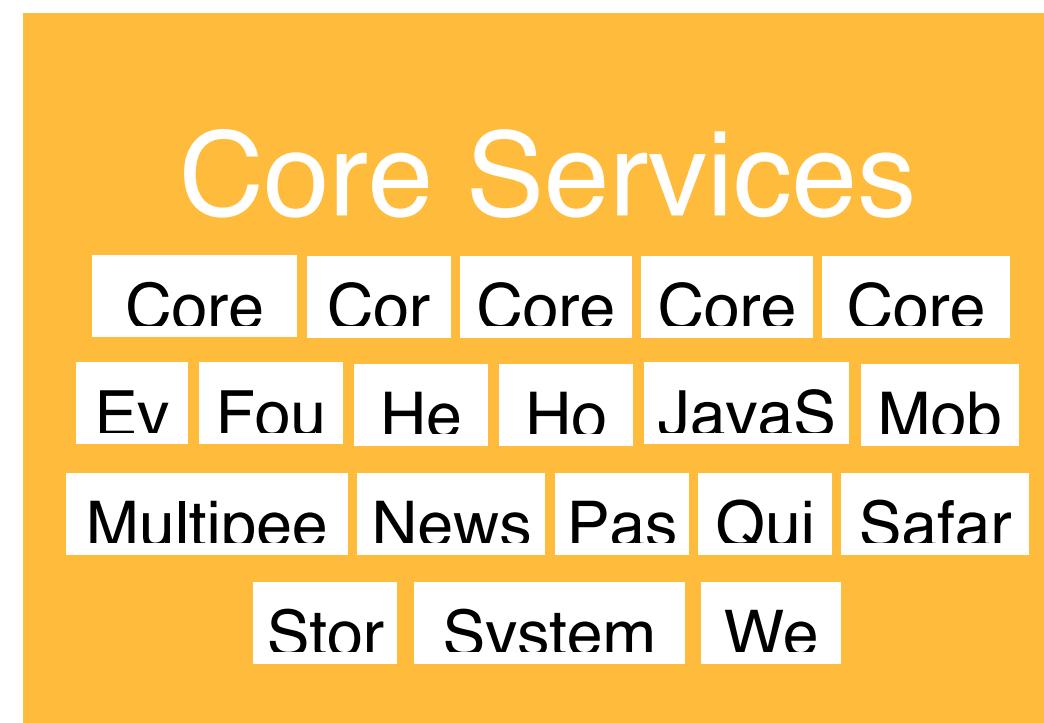


iOS STACK: CORE Services



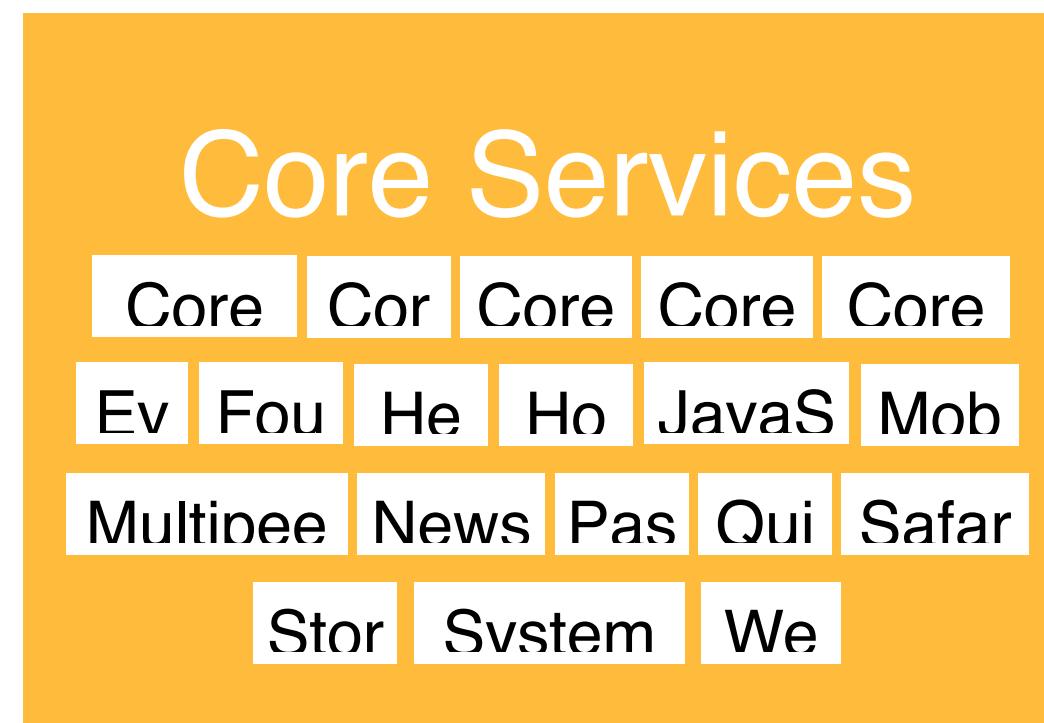
Access and manage key operating system services, such as launch and identity services.

Access to low-level routines, models, collections and data types, and wrappers.



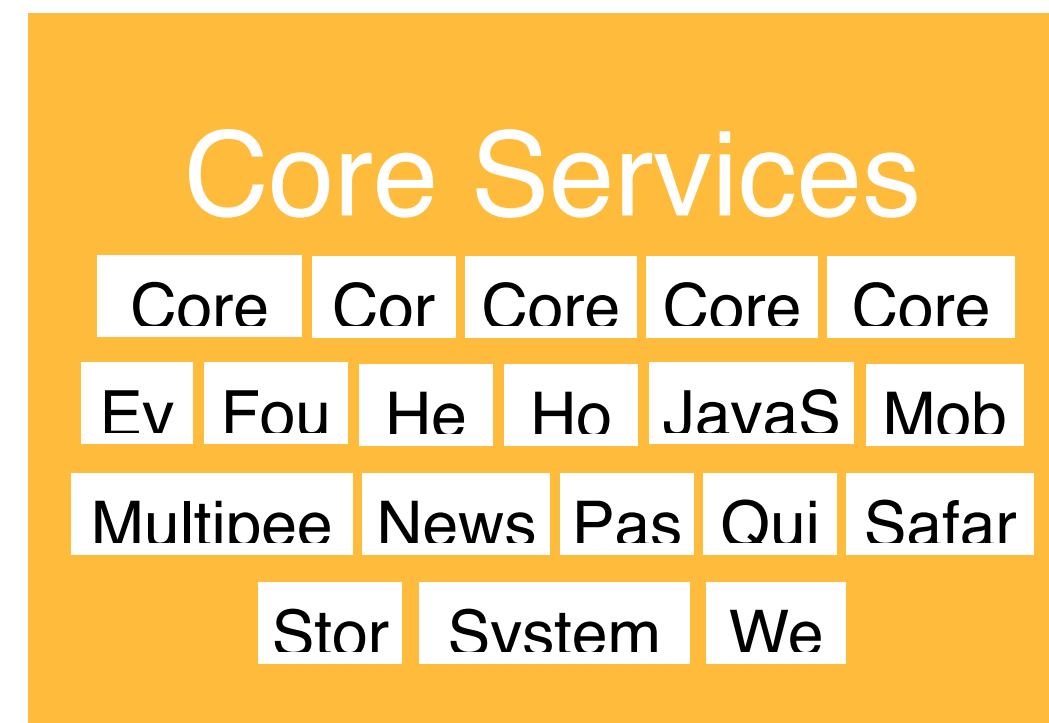
Core data (CoreData.framework):

- Model management in an app with MVC architecture.
- Core data creates, manages, and publish the data model.
- SQLite storage.



Core foundation (CoreFoundation.framework):

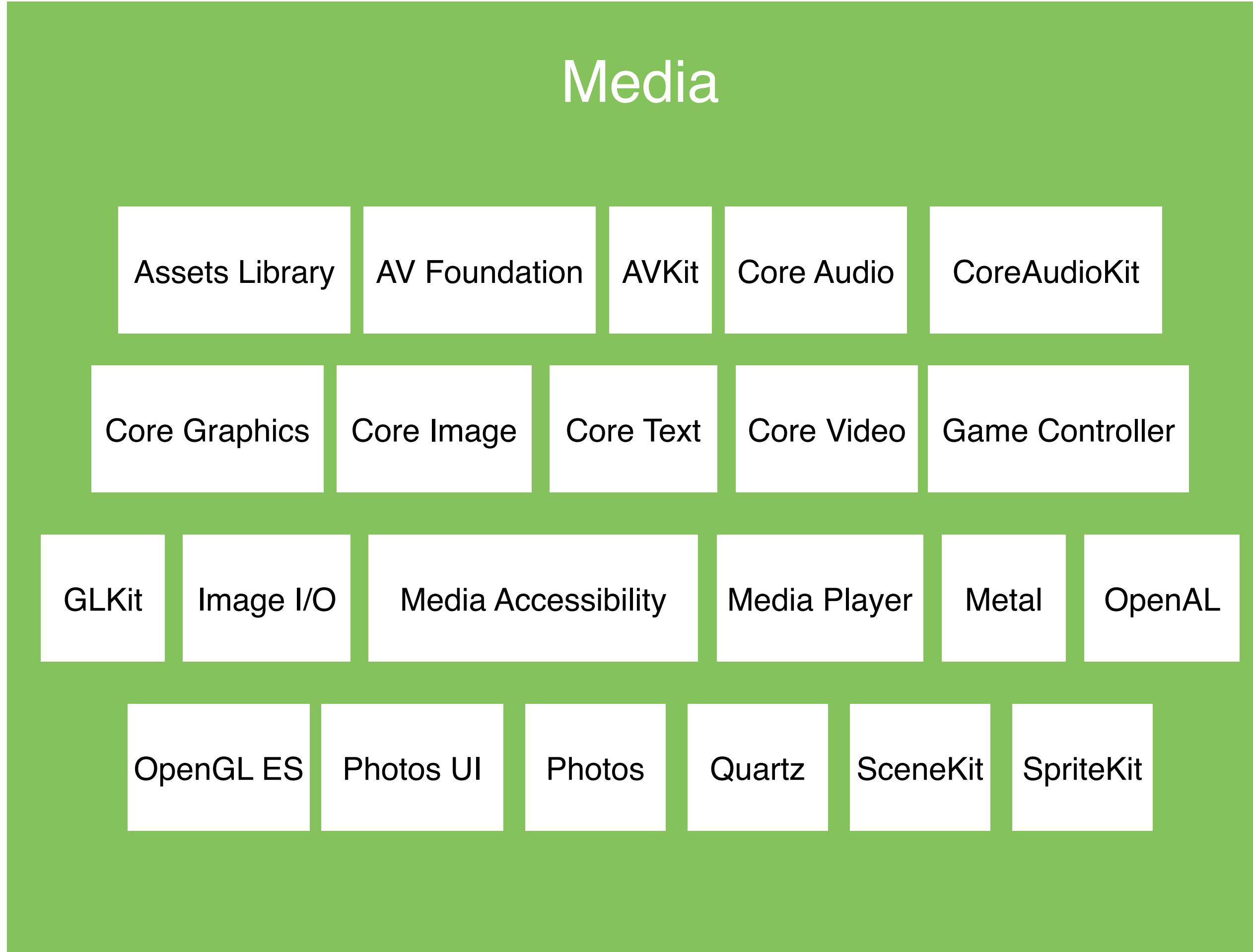
- Interfaces based on C.
- Bundles.
- String, Date, Time.
- Preferences.
- Threads.
- Ports/Sockets.
- URLs, Streams.



Foundation (Foundation.framework):

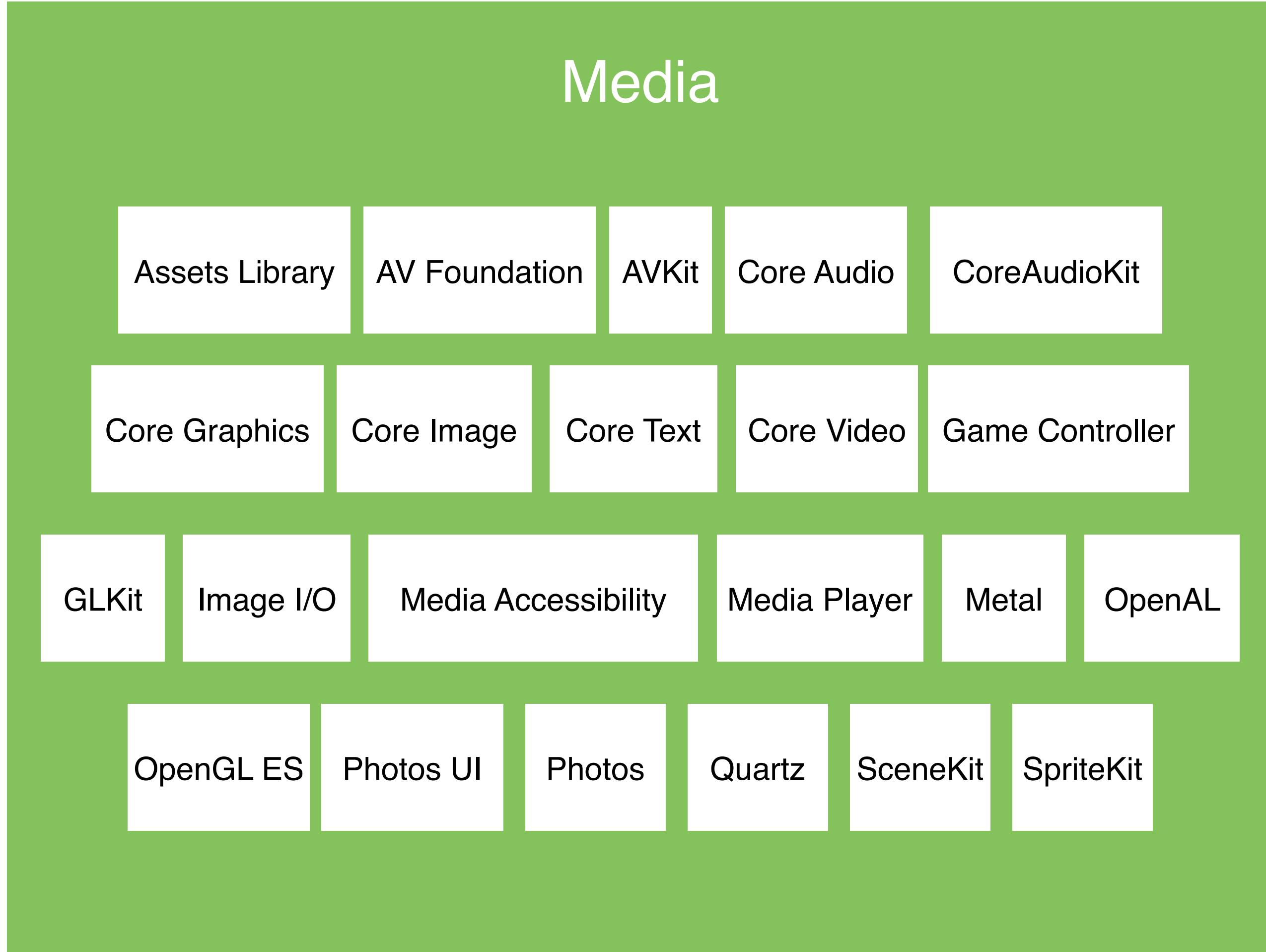
- Wrappers in Objective-C for interfaces in CoreFoundation.
- Same functionalities of CoreFoundation + Bonjour, Communication, Cache, Regexp.

iOS STACK: Media



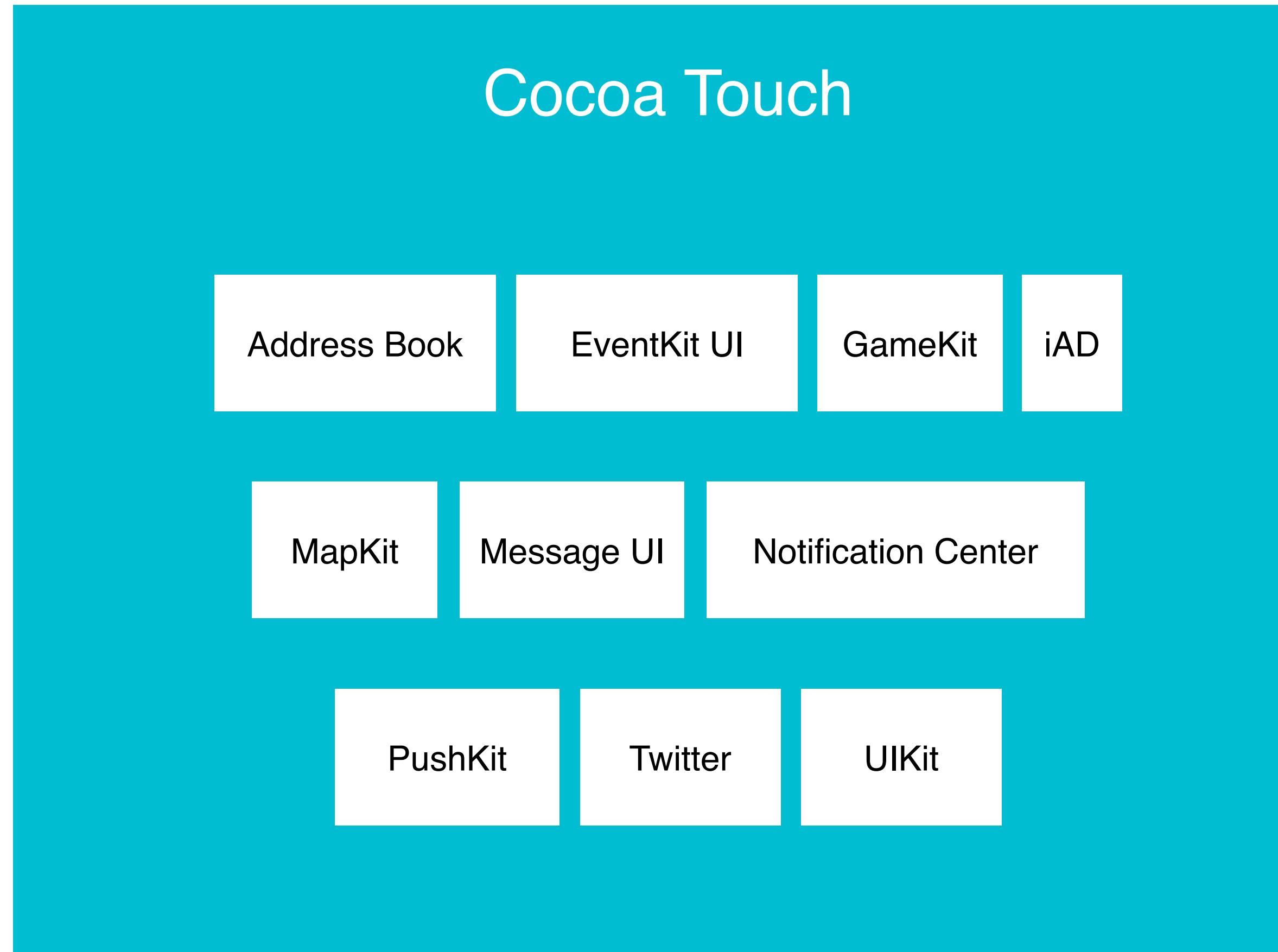
Frameworks for graphics, audio and video. Provide multimedia-related capabilities to iOS apps.

iOS STACK: Media



- Graphics, audio, and video technologies needed for multimedia experiences in the app.
- Graphic libraries: Quartz, OpenGL, Metal
- Audio libraries: Open AL, Core Audio, Media Player

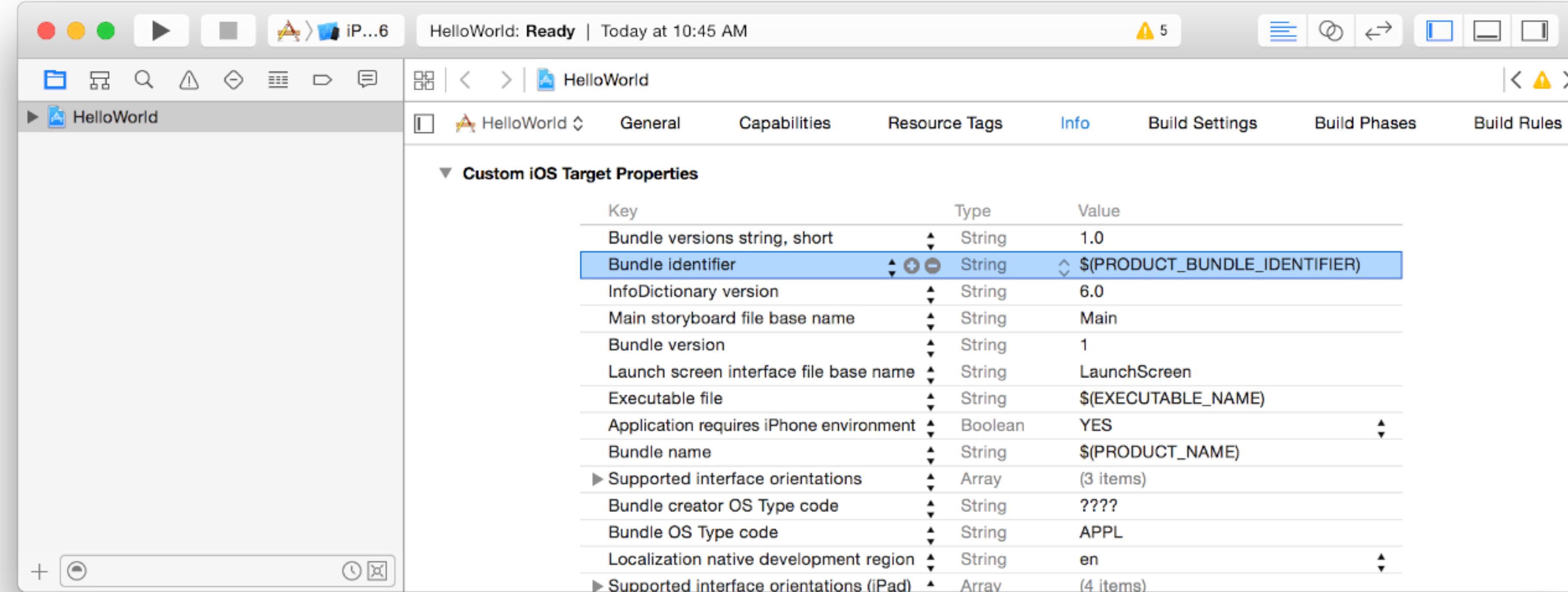
iOS STACK: Cocoa Touch



- Mainly written in Objective-C
- Frameworks to define the UI, Basic infrastructure, and key technologies to implement an app.
- Multi-tasking, touch-based input, notifications (push).
- Airdrop, contacts, camera.
- Text and typography.
- Storyboards.
- ViewControllers for standard interfaces

WHAT IS AN APP BUNDLE?

iOS STACK: Bundle



Distribution / packaging unit on MacOS and iOS

Folder with standardized hierarchical structure to group the resources of a project (executable, files, images, etc)

iOS STACK: Bundle

Bundle

MyApp.app

- Executable (Compiled code)
- Info.plist
- Icons
- Storyboards (layouts)
- Settings.bundle
- Resources

Distribution File

MyApp.ipa

- iTunesArtwork
- Payload → Bundles
- iTunesMetadata.plist

*ipa = iOS App Store Package