

WELCOME

Mobile Applications Testing



Platforms / OS :

iOS (iPhone) Architecture (simplified)

Cocoa Touch Layer

It is a top layer of the iPhone OS stack and it contains the frameworks that are most commonly used by iPhone application developers.

Media Layer

It is the second layer from the top of the stack. It provides the iPhone OS with audio, video, animation and graphics capabilities.

Core Services Layer

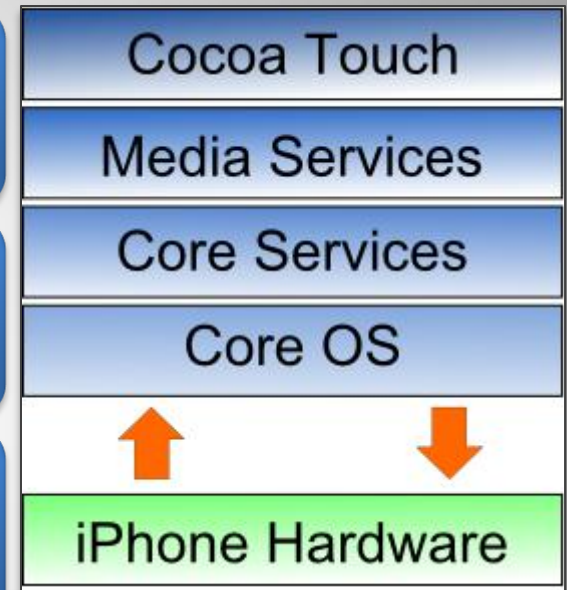
It is the third layer from the top of the stack. The iPhone Core Services layer provides much of the foundation on which the above layers are built.

Core OS Layer

The Core OS Layer is the bottom layer of the iPhone OS stack and sits directly on top of the device hardware. This layer provides a variety of services

iPhone Hardware

Hardware devices are managed by iPhone OS and provides the technologies needed for implementing native applications on the phone.




Platforms / OS :

iOS .IPA

- An **.ipa** file is an iOS application archive file which stores an iOS app.
- Each **.ipa** file includes a binary for the ARM architecture and can only be installed on an iOS device.
- Files with the **.ipa** extension can be uncompressed by changing the extension to **.zip** and unzipping.

Platforms / OS :

ANDROID vs IOS : SUMMARY

Category	Google Android	Apple iOS
License - Source	Open Source	Close - Proprietary
Company	Open Handset Alliance/Google	Apple
OS Platform	Linux	OS X – Unix (Darwin)
Programming Language	Java and Kotlin 	Objective-C and Swift-4
Virtual Machine	ART – Android Runtime (from KitKat 4.4 and Dalvik VM prior)	None
Package Manager	APK – Google Play/Play Store	IPA – APP Store
Carriers	AT&T, Verizon, T-Mobile, Sprint, MetroPCS, Virgin, 96% out of 369 US carriers	Apple Store, AT&T, Verizon, Sprint, Virgin Mobile, Cricket + 10 regional carriers
Life Battery	Moderate	Advance high
Browser Engine	WebKit: Web, Opera Mini, Dolphin, Chrome	WebKit: Safari, Opera Mini, Chrome

Mobile Ecosystem

Mobile World Statistics

Carriers/Service Providers

Network

Manufactures

Devices

Platforms/OS

Frameworks

API-Apps

Services

FRAMEWORKS : distinguishing features

SOFTWARE FRAMEWORK

Sets of libraries or classes

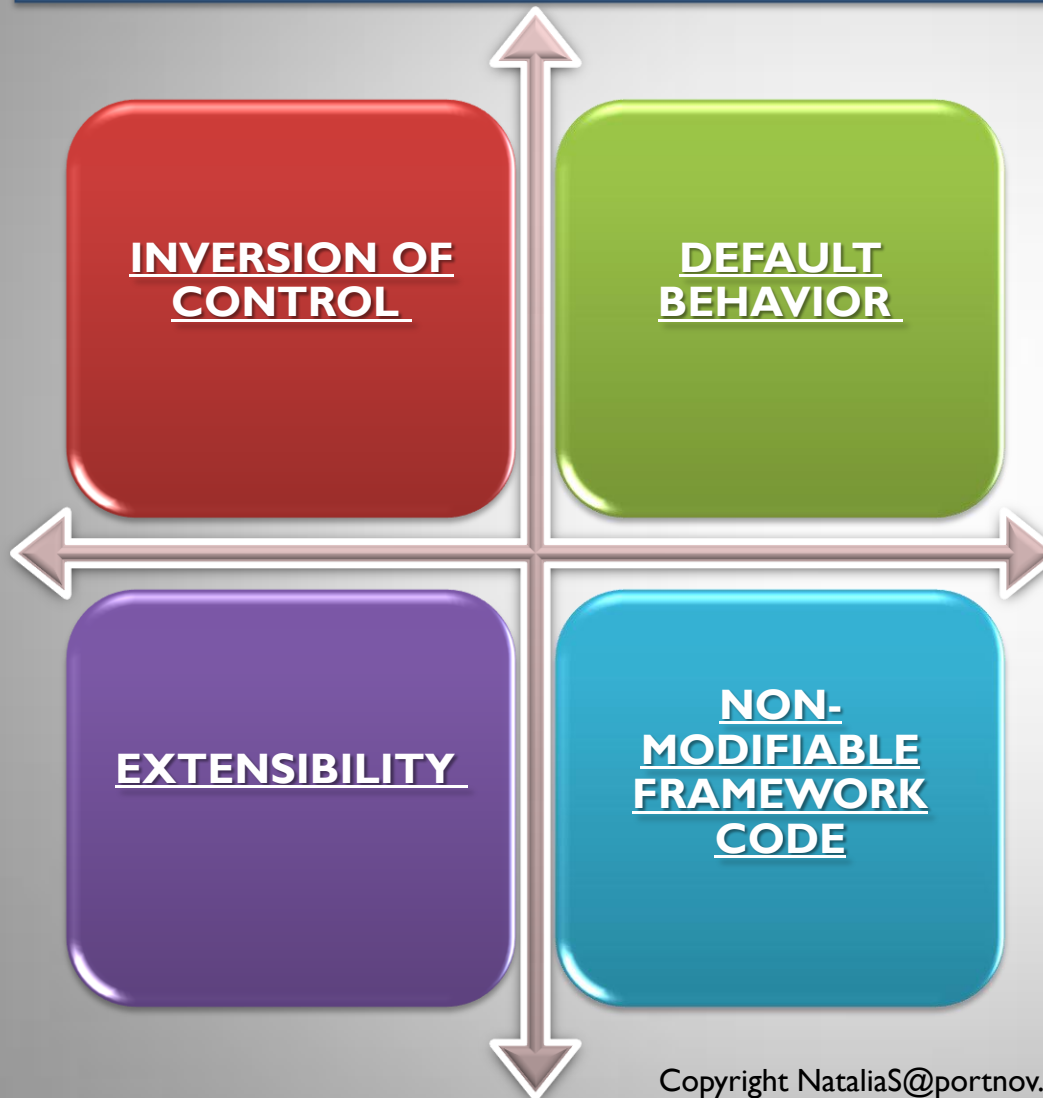
Built-in generic functionalities,
Deals with standard low-level details

Reusable software environment

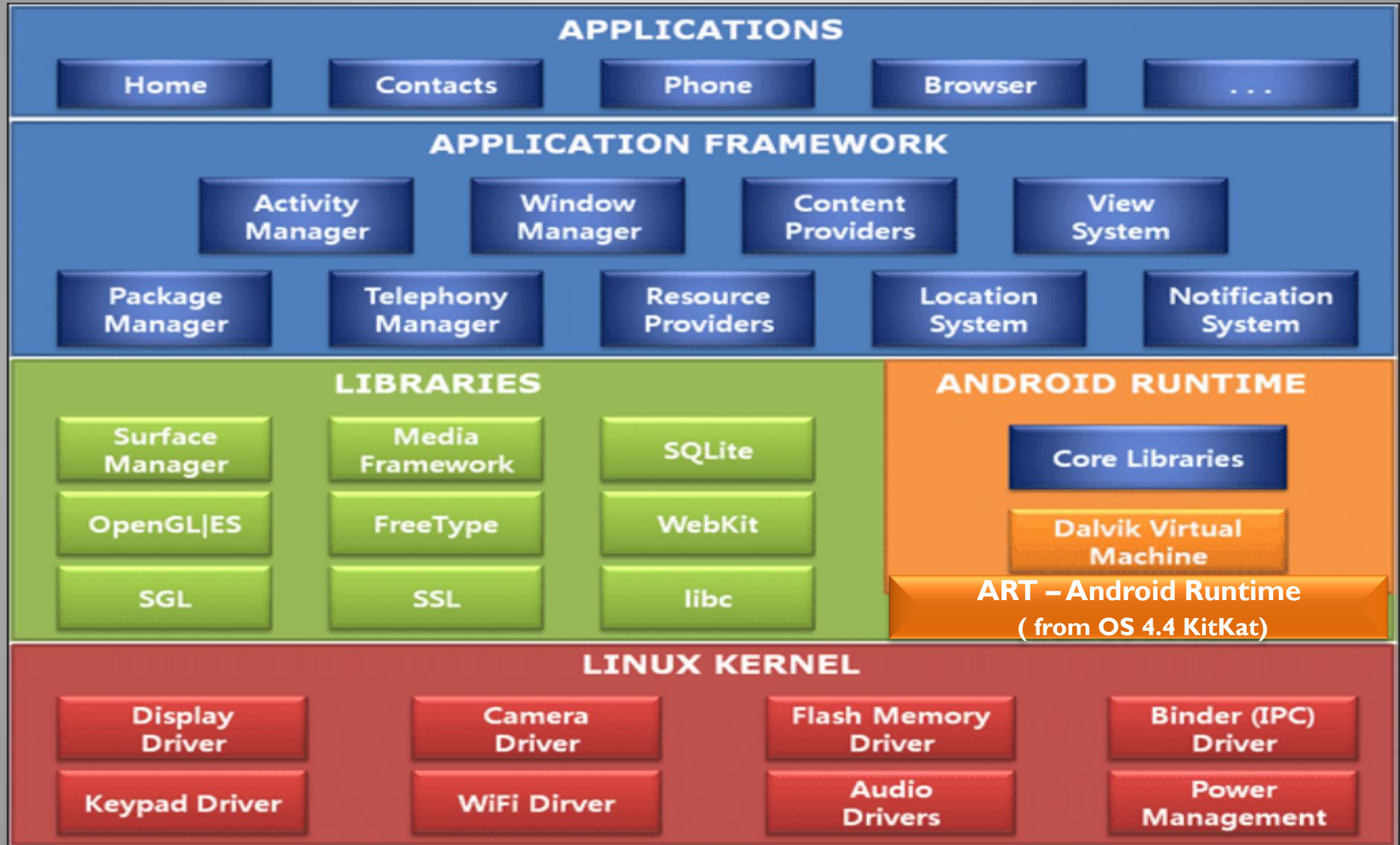
Working template application

Can be modified by writing additional code

FRAMEWORKS : distinguishing features



Platforms / OS : ANDROID ARCHITECTURE



FRAMEWORKS

Example → ANDROID APP

Application Framework sits on top of native libraries, android runtime and Linux kernel.

This framework come pre-installed with high-level building blocks that developers can use to program applications.

RIGHT SIDE → the most important application framework components for our application and Android development in general.

Activity Manager

• **Manages the lifecycle of application**

Content Provider

• **Stores and retrieves data and makes it accessible to all applications**

View system

• **Handles GUI related Tasks**

Package Manager

• **Retrieves various info related to the currently installed app on a device**

Resource Manager

• **Provides access to non-code resources such as icons, etc**

Location Manager

• **Location-based and related services**

Notification Manager

• **Executes and Manages all Notifications, alerts, etc**

FRAMEWORKS

Example → Win CE and OS X

SF is a universal, reusable software platform used to develop applications, products and solutions.

SF include support programs, compilers, code libraries ,APIs and tool sets that bring together all the different components to enable development of a project or solution.

- Platform: Windows CE □
- OS: Windows Phone
- Framework: .NET

- Platform: OS X □
- OS: iOS
- Framework: Cocoa Touch



FRAMEWORKS : CONCLUSION

SUMMARY

Soft development is about getting stuff done, not figuring out how to get it done.

Frameworks and libraries help the developers focus on creating rather than figuring stuff out.

Rather than reinventing the wheel, Developers can use a framework or library to delegate brunt, noncreative and repetitive work, freeing up their time and energy to create the actual website or application.



Mobile Ecosystem

Mobile World Statistics

Carriers/Service Providers

Network

Manufactures

Devices

Platforms/OS

Frameworks

API-Apps and other

Services

API-APP

An **Application Program Interface (API)** is code that allows software programs to communicate with each other.

It defines the correct way to write a program that requests services from an operating system or other application.

APIs are implemented by function calls.

The API defines the correct way for a developer to write a program that requests services from an Operating System or other application.



API-APP : How they Work?

APIs are made up of two related elements.

The first is a specification that describes how information is exchanged between programs, done in the form of a request for processing and a return of the necessary data.

The second is a software interface written to that specification and published in some way for use.

The software that wants to access the features and capabilities of the API is said to call it, and the software that creates the API is said to publish it.



API-APP Basic Types

APIs take three basic forms: local, web-like and program-like.

The three basic types of APIs

APIs take three basic forms: local, web-like and program-like. Here's a look at each type.

Local APIs

The original API, created to provide operating system or middleware services to application programs.

Web APIs

Designed to represent widely used resources like HTML pages and are accessed using a simple HTTP protocol. Often called REST APIs or RESTful APIs.

Program APIs

Based on RPC technology that makes a remote program component appear to be local to the rest of the software.

API-APP Basic Types

APIs take three basic forms:

Local APIs

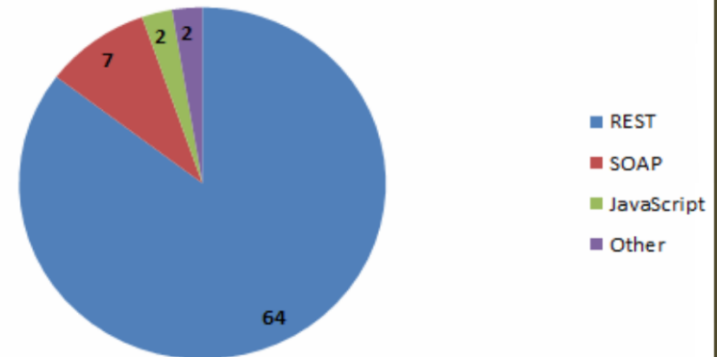
Local APIs are the original form, from which the name came.

They offer OS or middleware services to application programs.

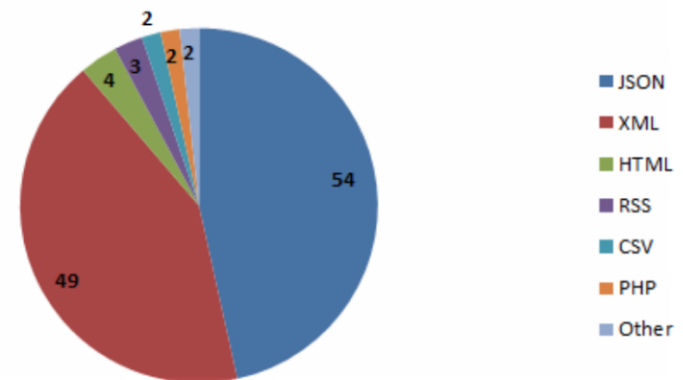
Example :

Microsoft's .NET APIs, the TAPI (Telephony API) for voice applications, and database access APIs

Protocols Used in Local APIs



Data Formats Used in Local APIs



API-APP Basic Types

Web APIs

APIs take three basic forms:

Web APIs are designed to represent widely used resources like [HTML](#) pages and are accessed using a simple [HTTP](#) protocol.

Any web [URL](#) activates a web API.

Web APIs are often called [REST](#) ([representational state transfer](#)) or [RESTful](#) because the publisher of REST interfaces doesn't save any data internally between requests.

As such, requests from many users can be intermingled as they would be on the internet.



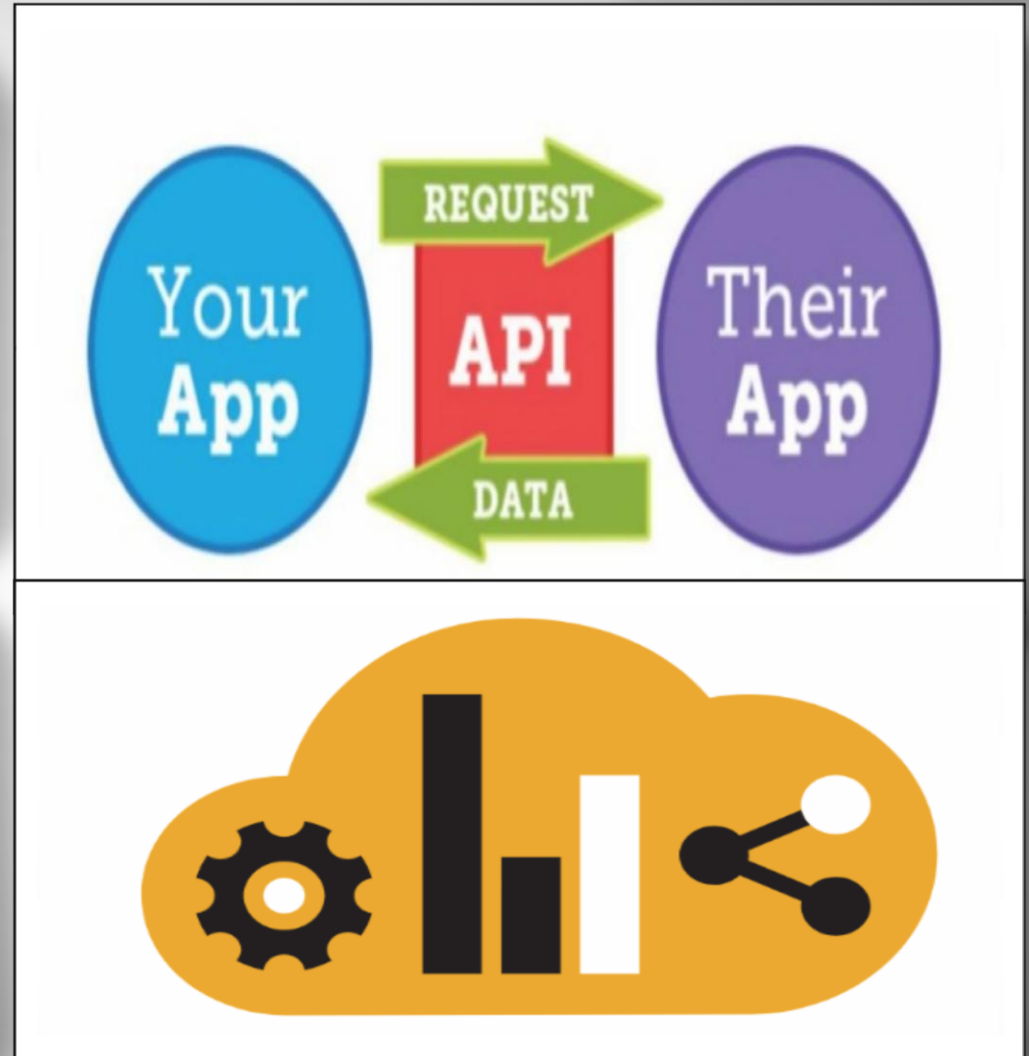
API-APP Basic Types

Program APIs

APIs take three basic forms:

Program APIs are based on remote procedure call (RPC) technology that makes a remote program component appear to be local to the rest of the software.

Service oriented architecture (SOA) APIs, such as Microsoft's WS-series of APIs, are program APIs.



API-APPS : EXAMPLES

The following list contains several examples of popular APIs:



Google Maps API: *Google Maps APIs lets developers embed Google Maps on webpages using a JavaScript or FlashInterface*



YouTube APIs: *YouTube API: Google's APIs lets developers integrate YouTube videos and functionality into websites or applications*



Flickr API: *The Flickr API is used by developers to access the Flickr photo sharing community dataThe Flickr API consists of a set of callable methods, and some API endpoints*



Twitter APIs: *Twitter offers two APIs. The REST API and Search API*



Amazon Product Advertising API: *Amazon's Product Advertising API gives developers access to product selection and discovery*



UserEngage : *Engagement tool, marketing automation platform that tracks you visitors*