

Introduction



ICT2105 Mobile Application Development

Jeannie S. Lee

Spring 2020

Overview

Popularity and growth of mobile device usage

Mobile device evolution

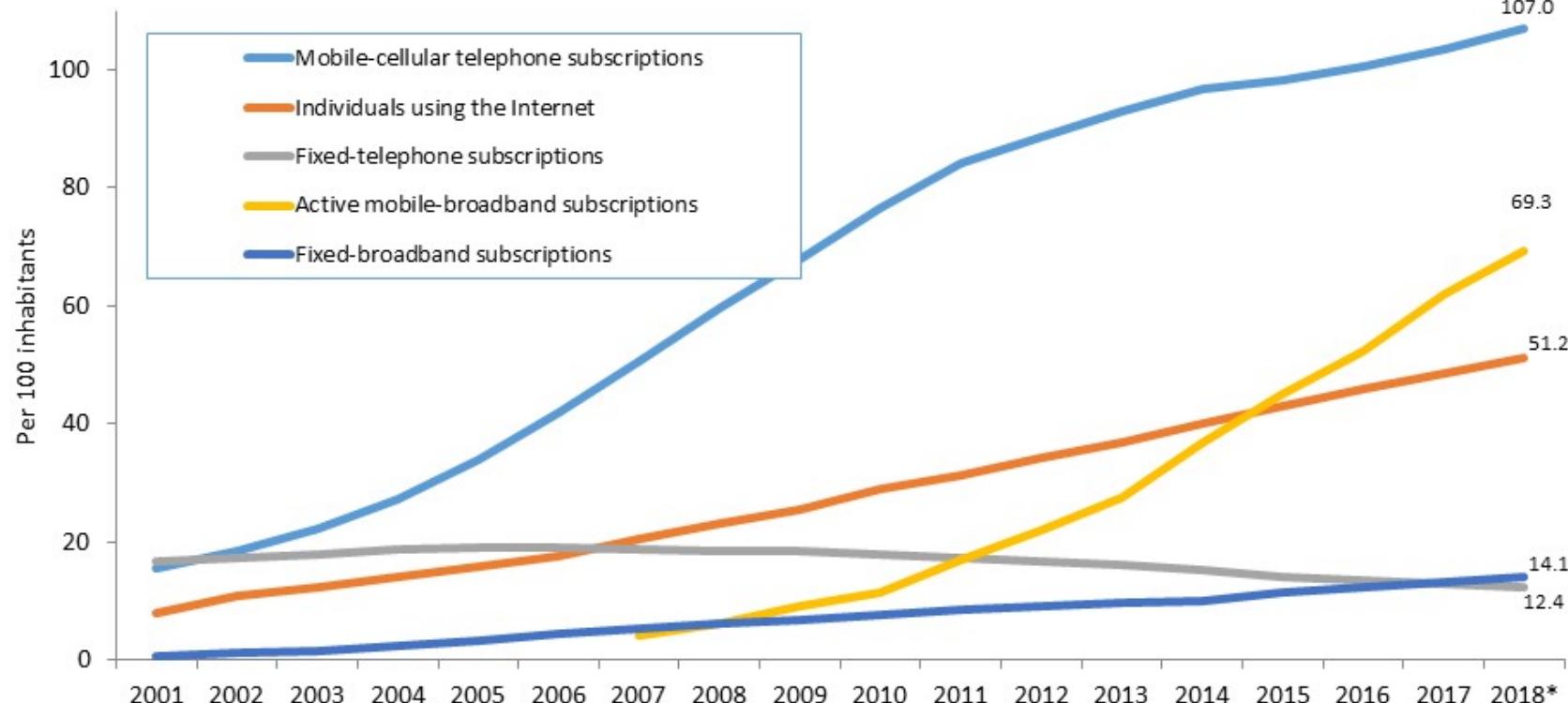
Operating Systems (Recap)

Mobile Hardware

Generic mobile architecture

Mobile platform overview

Global ICT developments, 2001-2018*



Note: * Estimate

Source: ITU World Telecommunication /ICT Indicators database

Mobile Device Popularity

Mobile usage is at an all-time high. U.S. consumers now dedicate **one-fifth** of their media consumption time to mobile. That's a share five times greater than in 2009.

Consumers are spending **as much time on mobile as they are in the traditional online category** (which includes all activity on desktops and laptops.)

Mobile was the **only media type to grow** in total U.S. consumer minutes spent per day from 2010 to 2012.

In the course of 2013, **tablet shipments have grown 83%** while PC shipments dropped 13%.

The 219 million mobile-only users now make up close to **20%** of Facebook's total user base

Facebook mobile advertising now represents 41% of its overall ad revenue.

Pinterest's U.S. mobile-only user base **grew 28%** reaching 18.3 million in June 2013.

Tablets and smartphones now account for **26%** of all local search traffic.



Motorola
8900X-2

Nokia
2146

Nokia
3210

Nokia
6210

Ericsson
T39

Alcatel
OT511

Samsung
E250

Apple
iPhone

BlackBerry
Curve 8900

Samsung
Galaxy S2

Samsung
Galaxy S4

Sony Xperia
Z Ultra

*Mainframe
Computing
1950's*



*Mini
Computing
1960's*



*Personal
Computing
1970's*



*Desktop Internet
Computing
1980's*



*Mobile Internet
Computing
1990's*



- Source: Morgan Stanley Report on Mobile Internet – an excellent read

Blast from the past





Last year's devices



Latest devices?



Mobile Device Evolution

Formerly proprietary OS and apps, sparse information available

Focused on single use-case, either calls or note taking

Rarely networked, no third-party application development

Improvement in hardware, battery and processing power and incorporation of many features e.g. GPS, camera, cellular modem, display

Third party application distribution service in 2008, Apple App Store

OS fragmentation and cross platform frameworks using HTML5 and javascript

Multiple use cases

Communication Device or Swiss Army Knife?



V.S.



Banking, PoGo, Whatsapp, txtng, selfies, maybe phone calls...

Mobile Resource Constraints

Mobile devices are resource-poor relative to static devices

Limited processing power and CPU

Limited RAM, disk, screen size, battery ...

Connectivity highly variable: Disconnection, low bandwidth, volatility

Location changes often

Higher security risk - More likely to be stolen/lost, many attack surfaces

M. Satyanarayanan, Fundamental Challenges in Mobile Computing (1995)

Mobile Contexts

- Location changes often
- Users subject to disturbances
- Lower attention span
- Intermittent interaction
- No full concentration



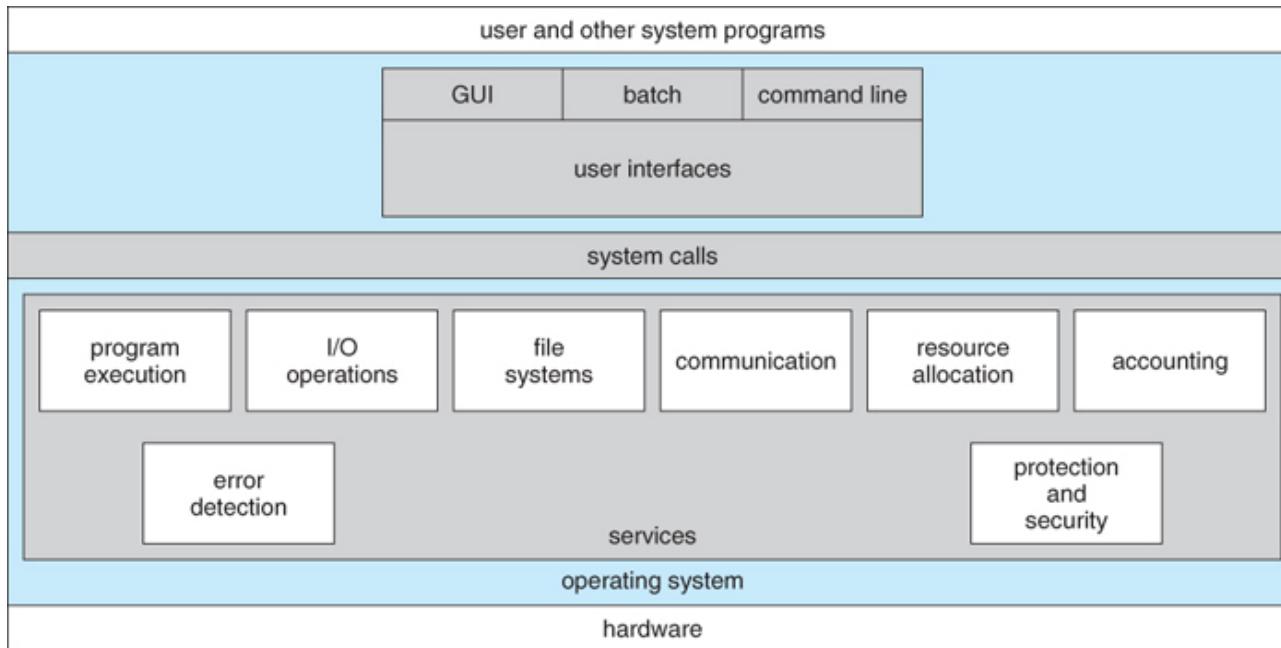
Operating System (Recap)

A program that acts as an intermediary between a user of a computer and the computer hardware

The goals of an operating system are to execute and control user programs, and to manage hardware (and other) resources

OSes provide environments in which programs run, and services for the users of the system

Operating System Services (Recap)



View of operating system services (Silberschatz, Gagne, Galvin)

Hardware



Mobile processors usually based on the ARM RISC architecture

32 & 64-bit processors



Lower cost, heat and power consumption



Chip designs are licensed

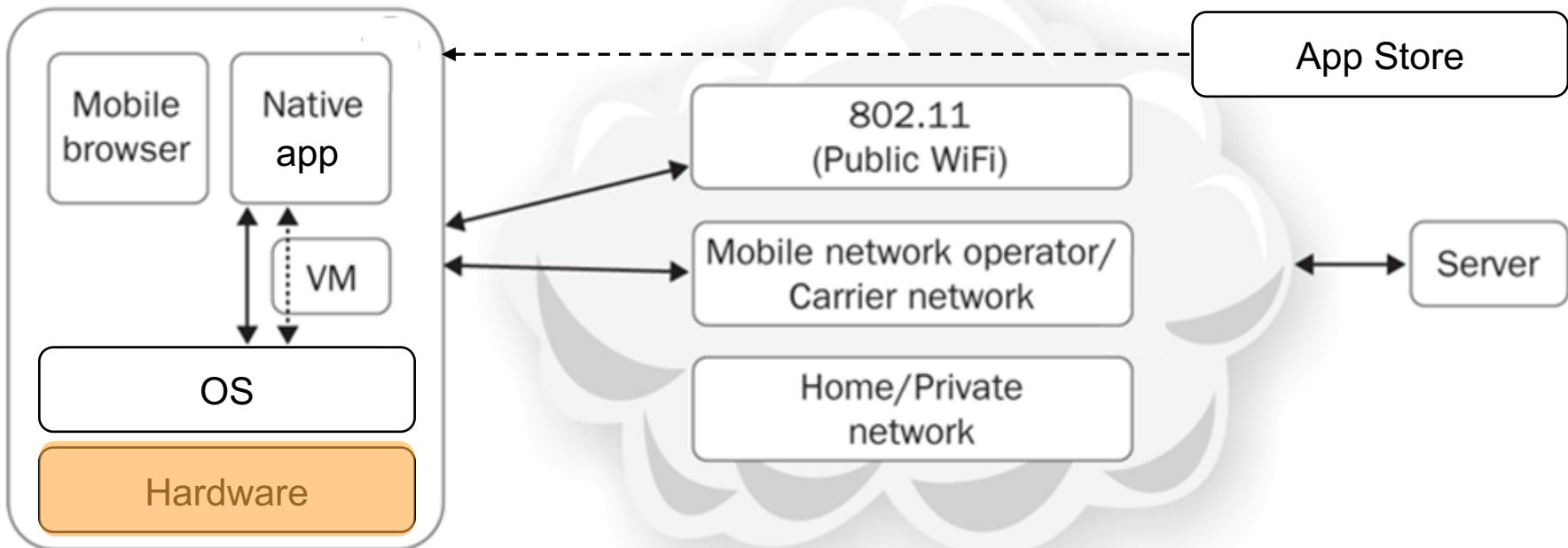


Some companies that make chips that implement an ARM architecture:

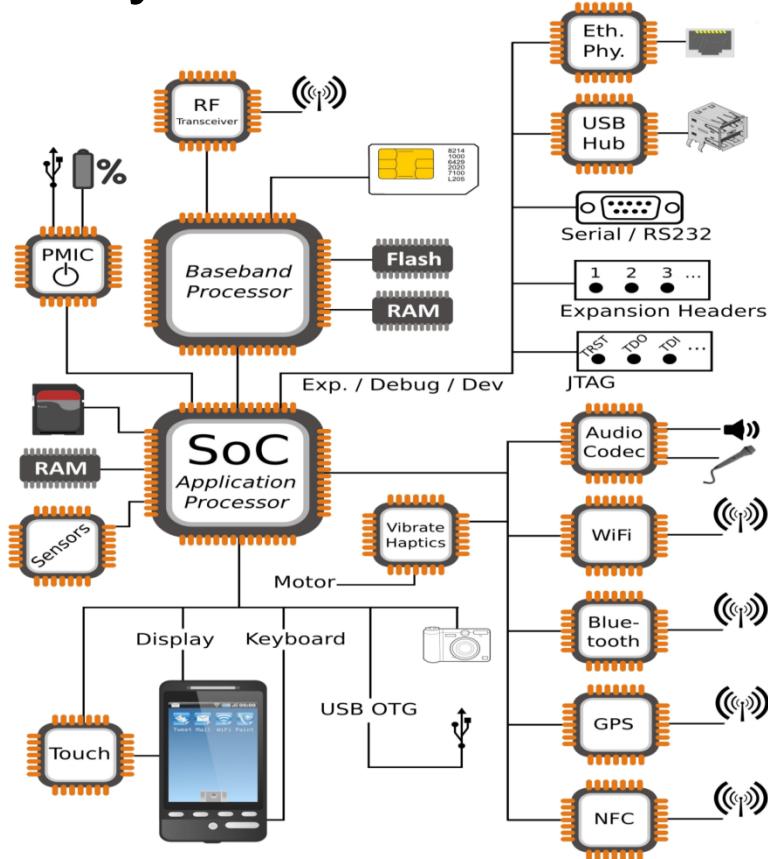
Apple, Analog Devices, Broadcom, Qualcomm, Samsung, ST Microelectronics, Texas Instruments

There are CISC architecture mobile devices too!

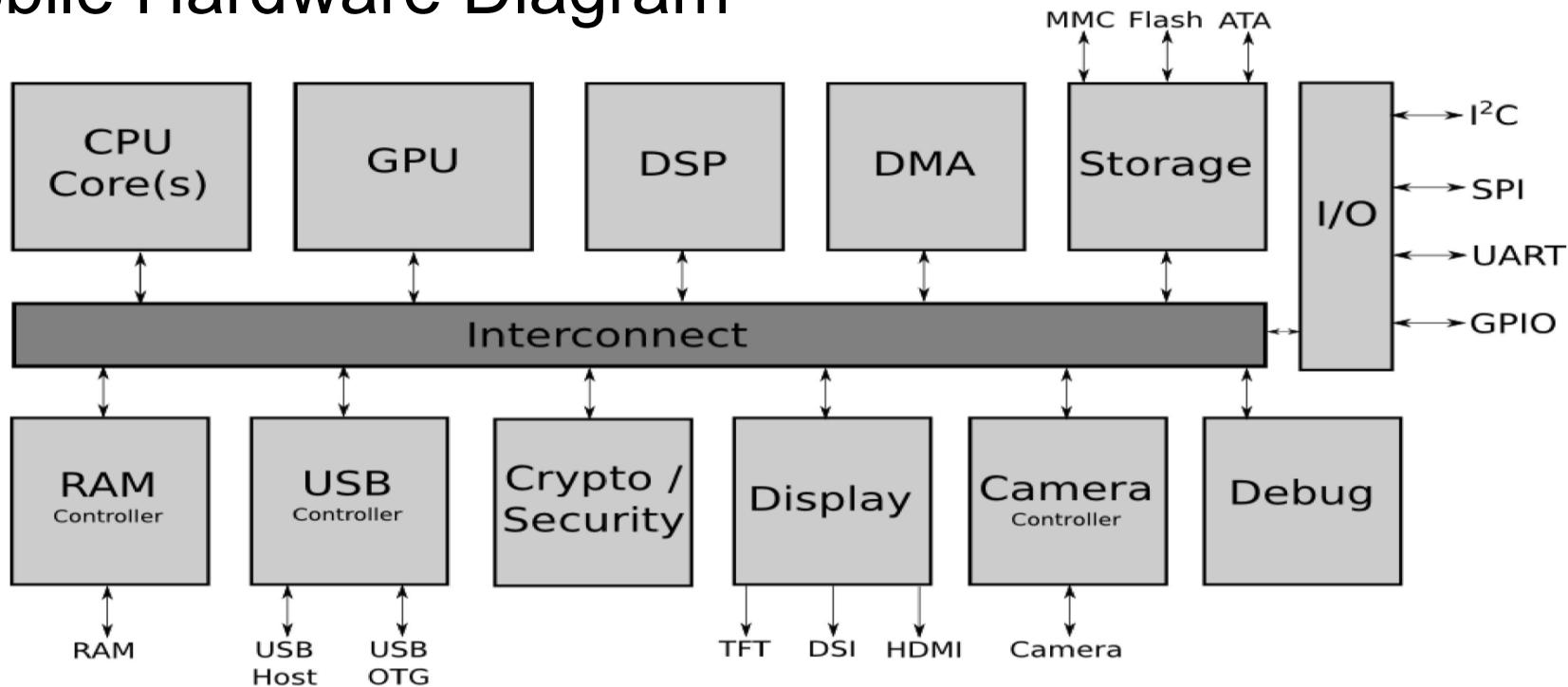
Generic Mobile Architecture



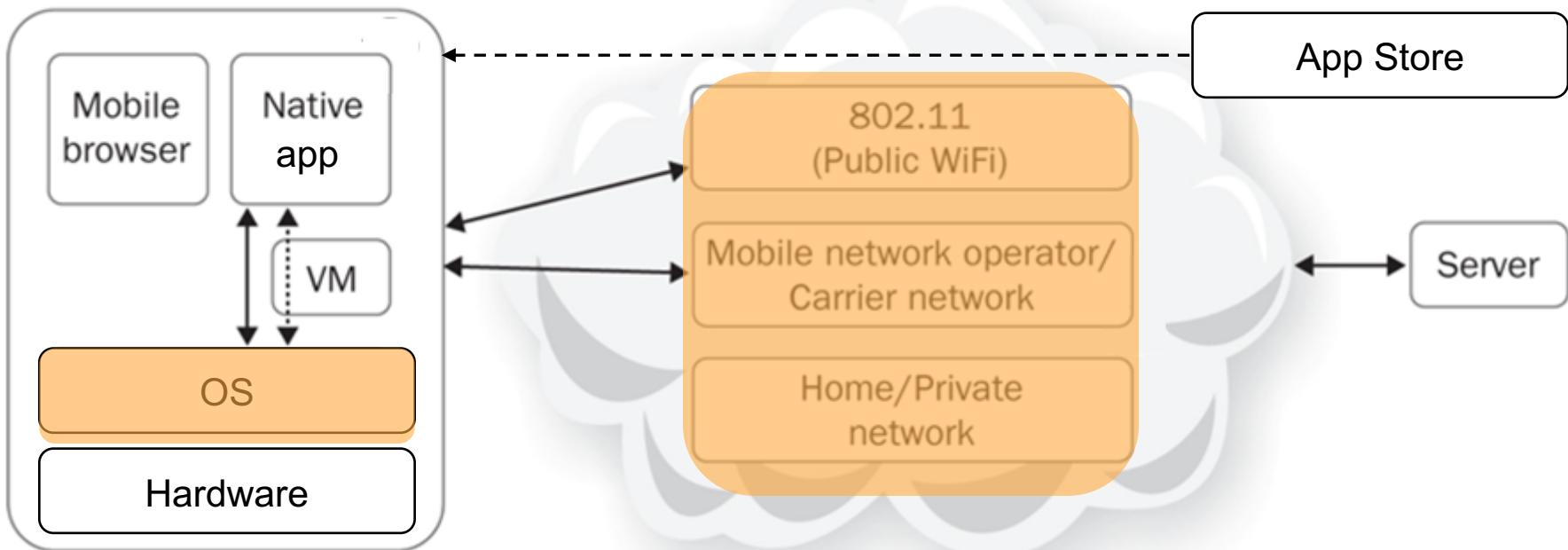
Mobile Phone Physical Hardware



Mobile Hardware Diagram



Generic Mobile Architecture



Generic Mobile Architecture (1)

Operating System

Manages computer hardware and software resources and provides common services for computer programs

Provides environments in which programs run, and services for the users of the system E.g. libraries, file system access, IPC, system calls

Closed or open source (Compare Android vs iOS, Windows Phone)



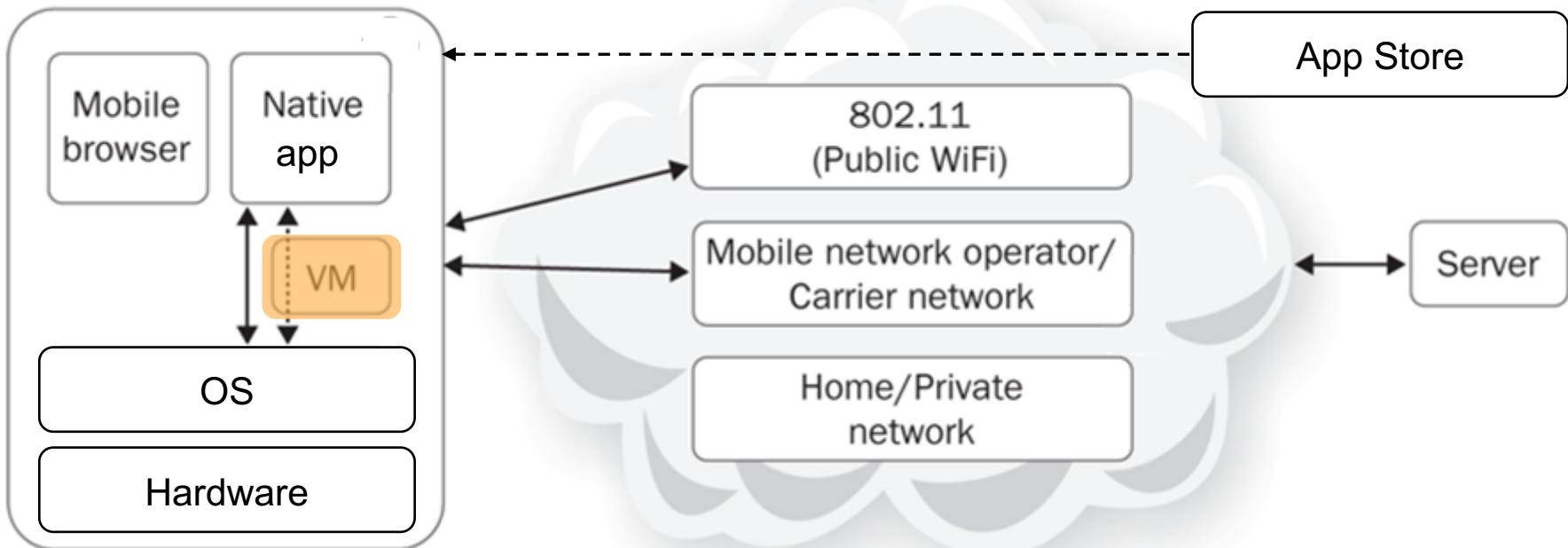
Network Access

Telco provider's cellular network (3G/4G/LTE/5G) or WiFi for Internet access

Connections may be intermittent or spotty

Public WiFi often used

Generic Mobile Architecture



Generic Mobile Architecture (2)

Virtual Machine (VM)

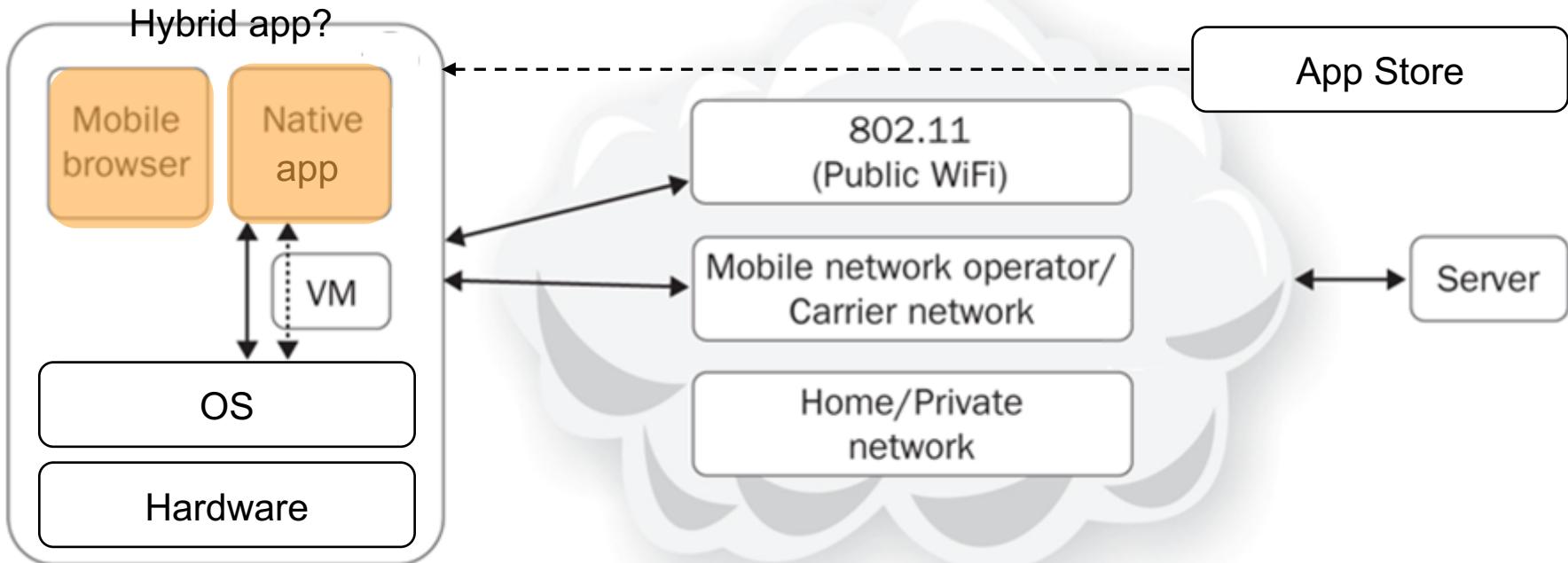
Abstracts the hardware of a single computer into several different execution environments

Executes bytecode (intermediate instruction set for execution by the VM)

Java runs in the Java Virtual Machine (JVM)

C# runs in Common Language Runtime (CLR)/Windows RunTime (WinRT)

Generic Mobile Architecture



Generic Mobile Architecture (3)

App

A self-contained program or piece of software designed to fulfil a particular purpose; an application, especially as downloaded by a user to a mobile device, typically associated with smart phone and tablets (*Oxford dict definition*)

Browser

Web browser that is optimized to display Web content on smaller screens

Optimized for low memory and low bandwidth of mobile devices

Also an app by itself

Generic Mobile Architecture (4)

Native Apps

Applications that directly execute in the OS, and is supported with minimal computational overhead and additional components

Native applications may execute with or without VM or sandbox

- Objective-C runs directly in iOS
- Languages like Java operate in VM, runs in a sandbox

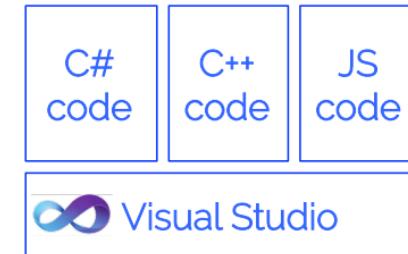
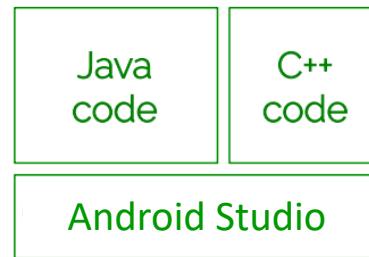
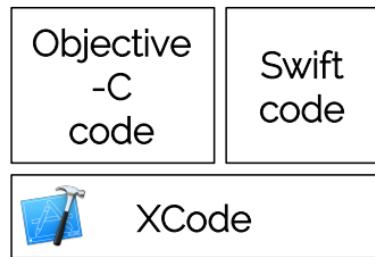
Have access to all device features E.g. camera, GPS, accelerometer, contacts

Usually installed through the app store

Written in the programming language specific for the platform

Many different platforms out there (fragmentation)

Native Application Development



Ivano Malavolta, End Users' Perception of Hybrid Mobile Apps, 2015

Generic Mobile Architecture (5)

Hybrid Apps

Developed using standard web technologies (HTML5, CSS and JavaScript)

Run inside a native container, and leverage the device's browser engine (but not the browser) to render the HTML and process the JavaScript locally.

May not have access to full device features

Ionic, Cordova, PhoneGap, Appcelerator, Sencha, Adobe Air, etc.

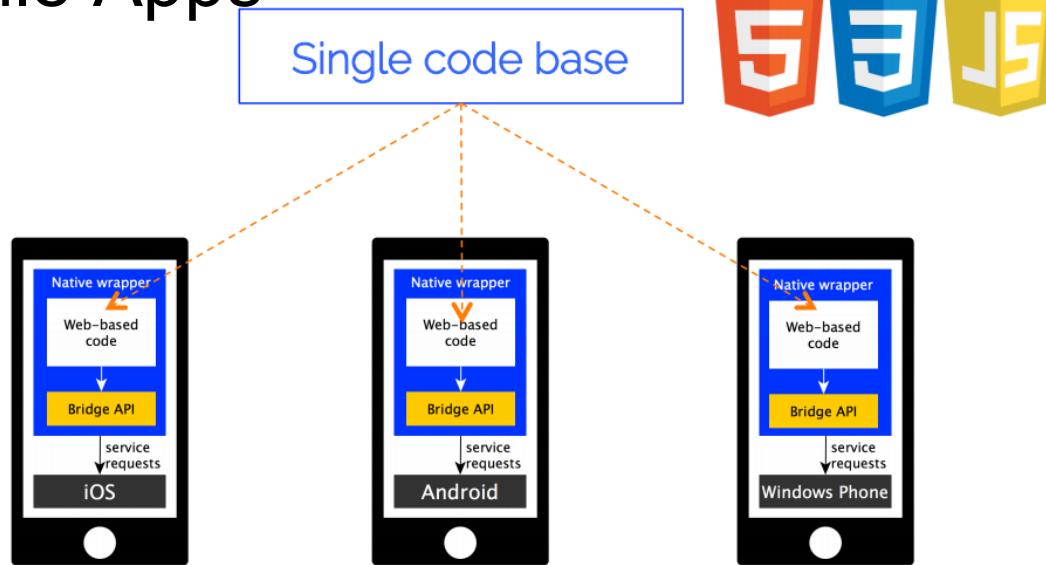
Web Apps

Run in the mobile browser, may need web server

Written in web programming languages (HTML, JavaScript)

Broadest audience, compatible with most devices

Hybrid Mobile Apps



- apps are developed using **standard web technologies**
- on top of a hybrid development framework
 - providing a **native wrapper** and a generic JavaScript API that **bridges** all the service requests to the corresponding platform API

Ivano Malavolta, End Users' Perception of Hybrid Mobile Apps, 2015

Hybrid App Pros & Cons

Pros

Cross-platform portability

Reuse of existing knowledge of web developers

Simpler and less expensive development process

Cons

Restricted access to hardware features

Decrease in performance

Variations on user experience

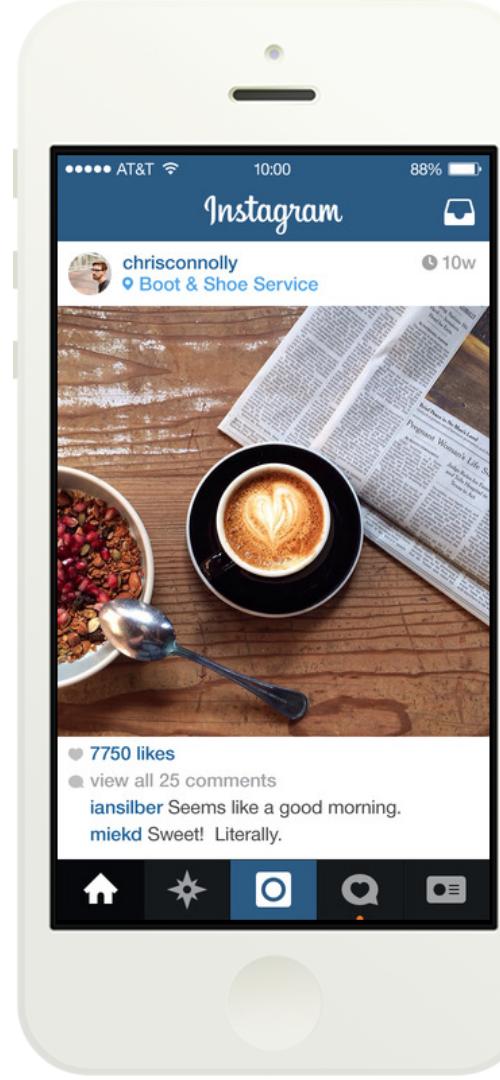
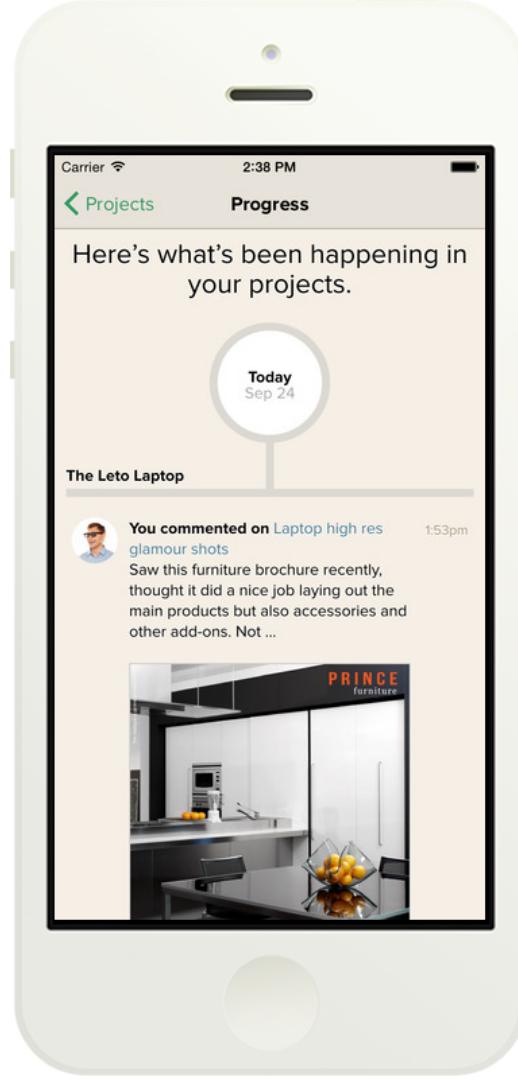
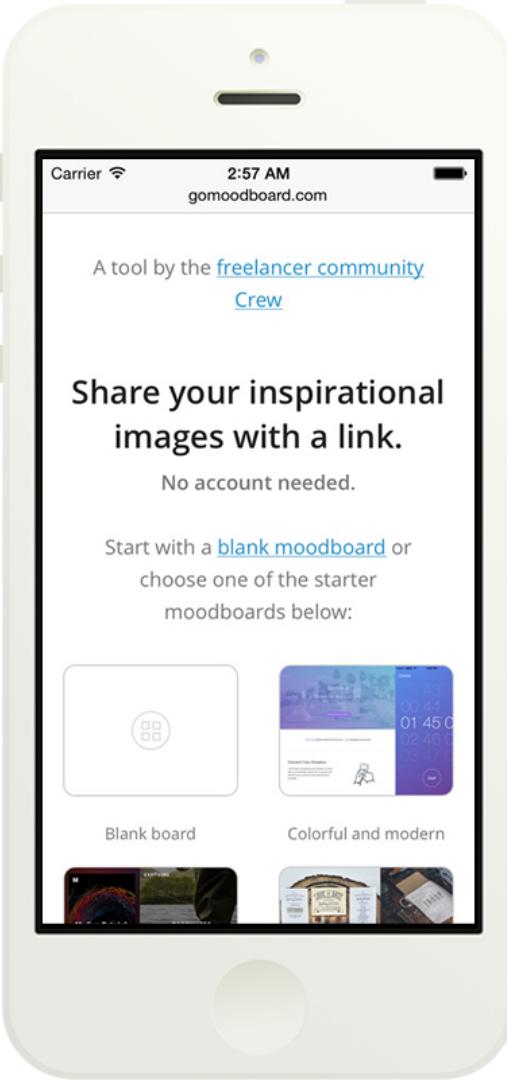
NATIVE vs. WEB vs. HYBRID: 7 FACTORS OF COMPARISON

KEY CON PRO NEUTRAL

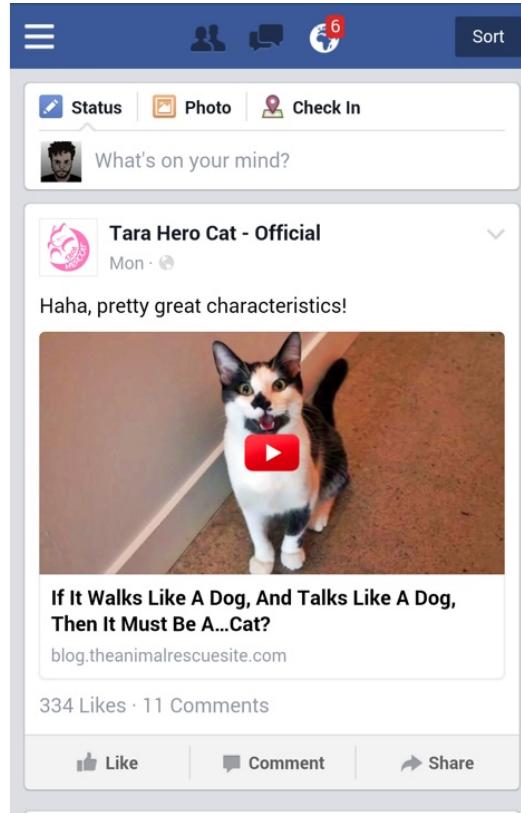
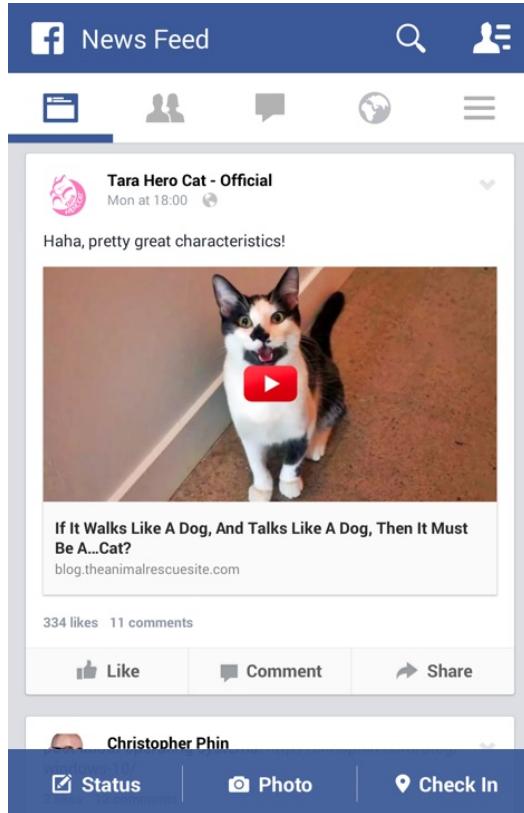
	NATIVE	HYBRID	WEB
COST	Commonly the highest of the three choices if developing for multiple platforms	Similar to pure web costs, but extra skills are required for hybrid tools	Lowest cost due to single codebase and common skillset
CODE REUSABILITY/PORTABILITY	Code for one platform only works for that platform	Most hybrid tools will enable portability of a single codebase to the major mobile platforms	Browser compatibility and performance are the only concerns
DEVICE ACCESS	Platform SDK enables access to all device APIs	Many device APIs closed to web apps can be accessed, depending on the tool	Only a few device APIs like geolocation can be accessed, but the number is growing
UI CONSISTENCY	Platform comes with familiar, original UI components	UI frameworks can achieve a fairly native look	UI frameworks can achieve a fairly native look
DISTRIBUTION	App stores provide marketing benefits, but also have requirements and restrictions	App stores provide marketing benefits, but also have requirements and restrictions	No restrictions to launch, but there are no app store benefits
PERFORMANCE	Native code has direct access to platform functionality, resulting in better performance	For complex apps, the abstraction layers often prevent native-like performance	Performance is based on browser and network connection
MONETIZATION	More monetization opportunities, but stores take a percentage	More monetization opportunities, but stores take a percentage	No store commissions or setup costs, but there are few monetization methods

Dzone, The State of Native vs. Web vs. Hybrid (2014)

<https://dzone.com/articles/state-native-vs-web-vs-hybrid>



Facebook App, Native or Hybrid?



React Native

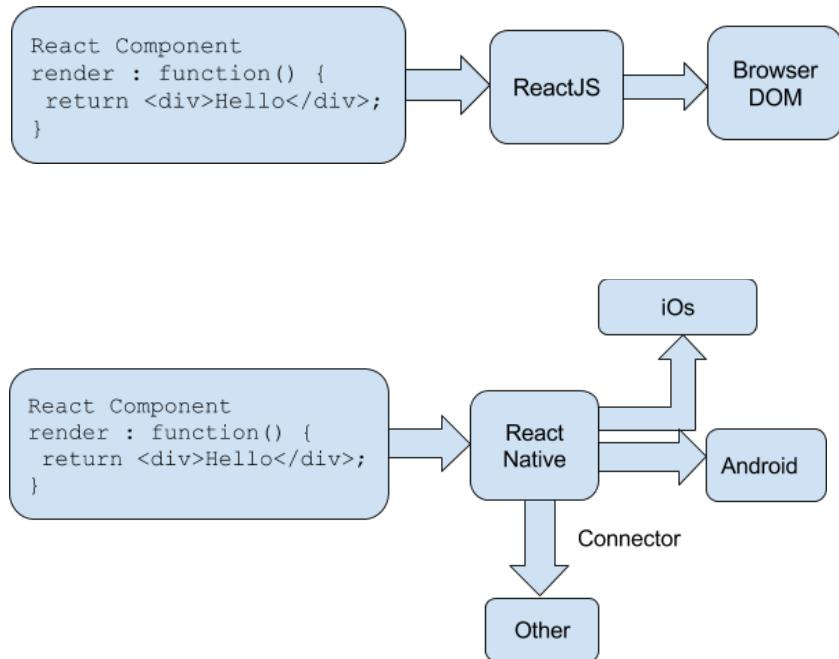
Use JavaScript and the React library to write mobile apps

JavaScript wrapper for native UI widgets

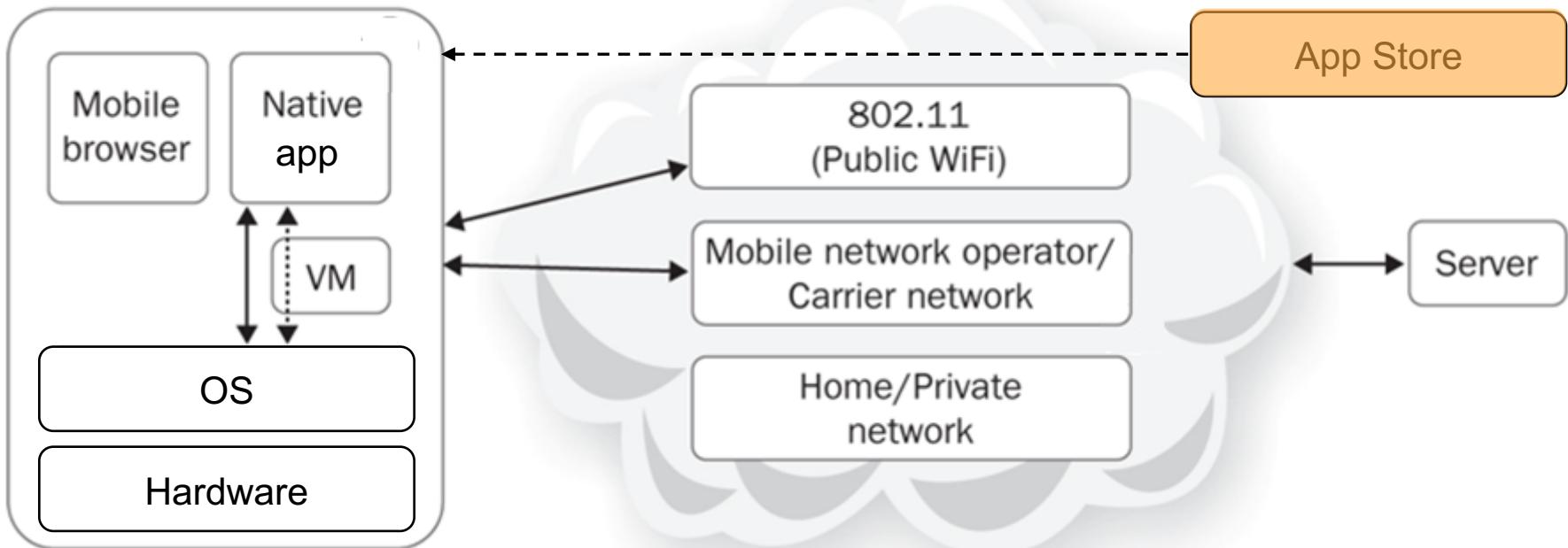
Invokes Objective-C APIs to render iOS components, or Java APIs to render Android components.

Open source project by Facebook
<https://facebook.github.io/react-native/>

Source Code:
<https://github.com/facebook/react-native>



Generic Mobile Architecture



Generic Mobile Architecture (6)

App Stores

Application distribution service, central place to obtain apps

May be third-party

Verizon Vcast (~2001)

Apple App Store (June 2008)

Android Marketplace, now Google Play Store (Oct 2008)

Fragmentation

Many different mobile platforms out there

A native mobile application is written from scratch for each platform

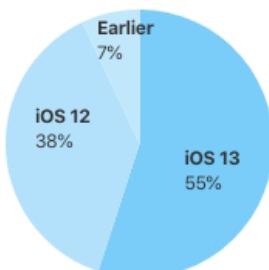
Main mobile platforms (2020): Android, iOS

(Sorry, Windows Phone)

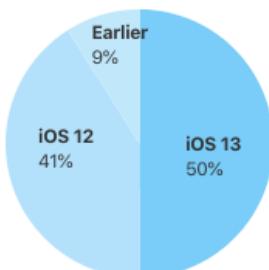


iPhone

55% of all devices introduced in the last four years use iOS 13.



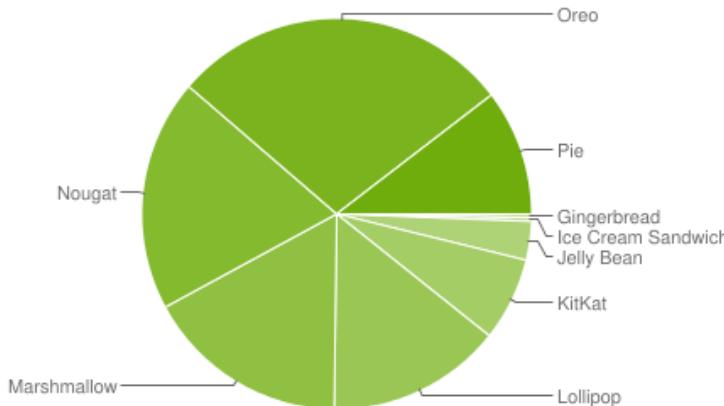
50% of all devices use iOS 13.



Android Fragmentation 2019/2020

Huge problem for Android. Some devices are no longer updated.

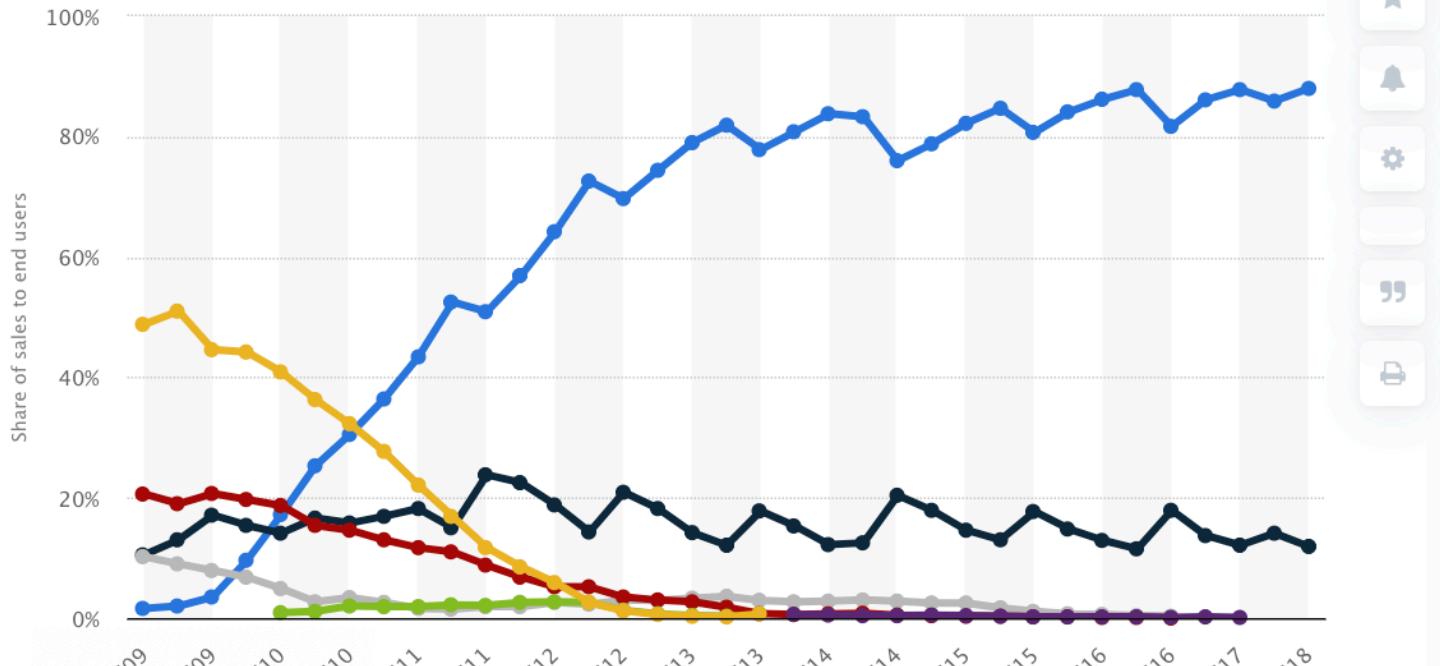
Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%



Data collected during a 7-day period ending on May 7, 2019.
Any versions with less than 0.1% distribution are not shown.

As measured by the App Store on October 15, 2019.

Global Mobile OS Market Share 2009-2018



iOS

OS that runs on Apple devices like iPhone, iPod, iPad & Apple TV

Proprietary operating system

Originated from Darwin and code from NeXTSTEP and BSD

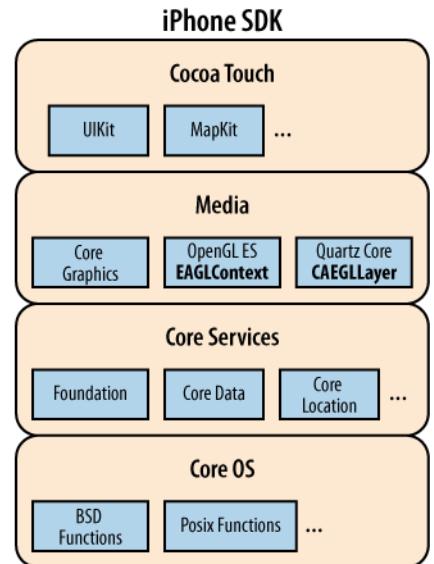
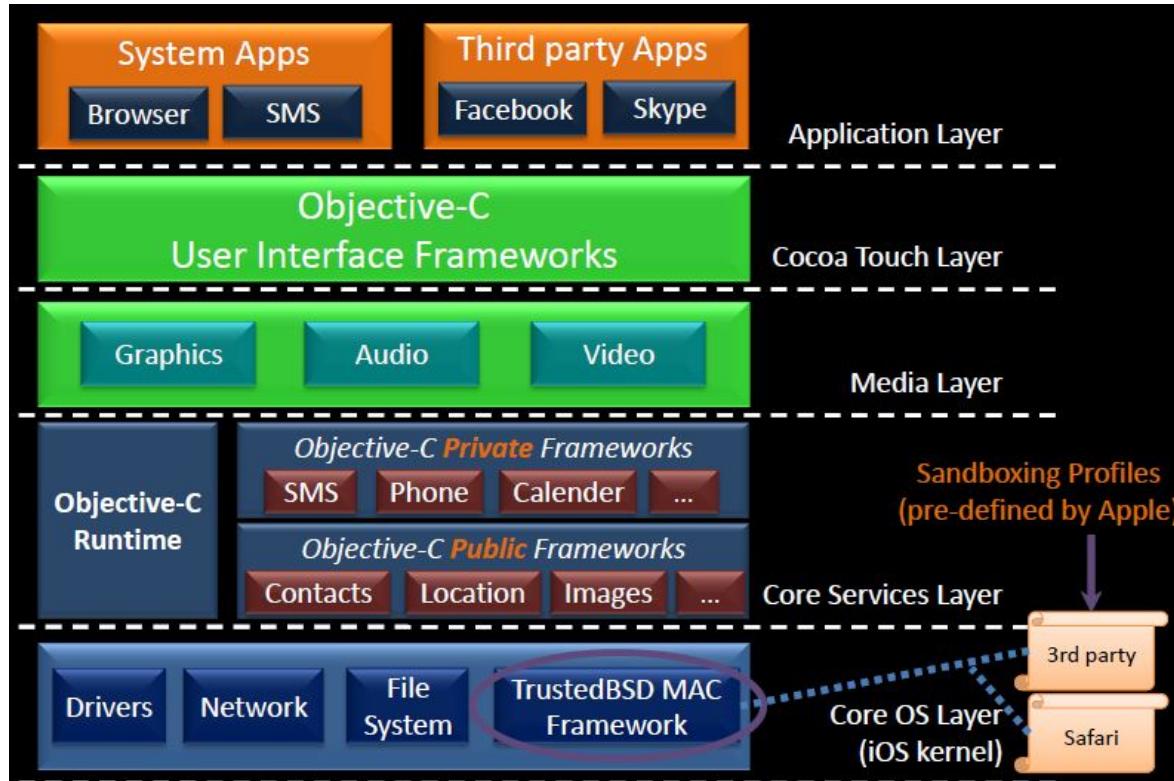
Stripped down Mac OS X + XNU kernel (Darwin's kernel)



iOS shares with OS X some frameworks such as Core Foundation and Foundation Kit, But UI toolkit is Cocoa Touch

Only allows to run Apple signed applications

iOS Software Stack & SDK



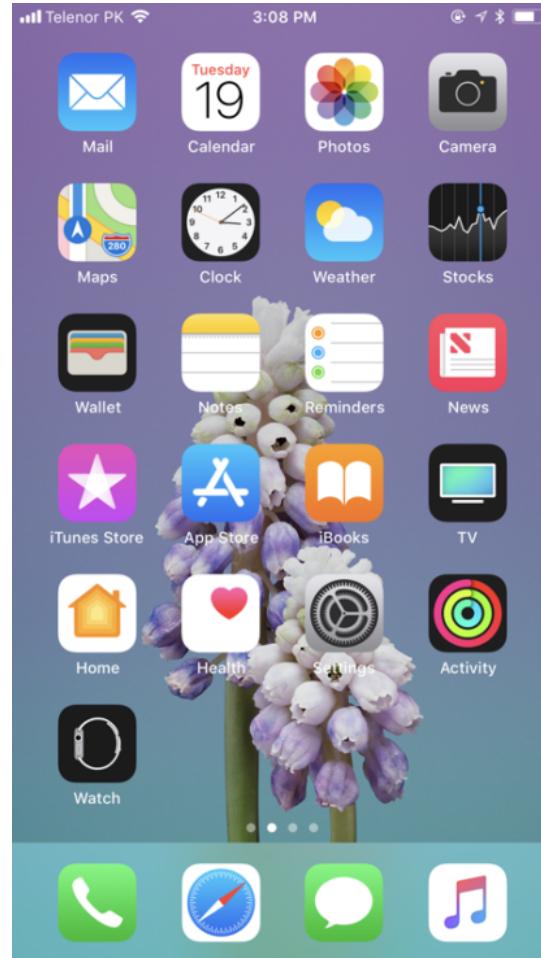
iOS

Languages: Objective-C, Swift, C/C++

Compile for iOS using the iOS SDK for 64
or 32bit ARM architecture

IDE: Xcode

Apple App Store



Android

OS developed by Google, open sourced (AOSP)

Based on the Linux kernel, middleware and libraries written in C

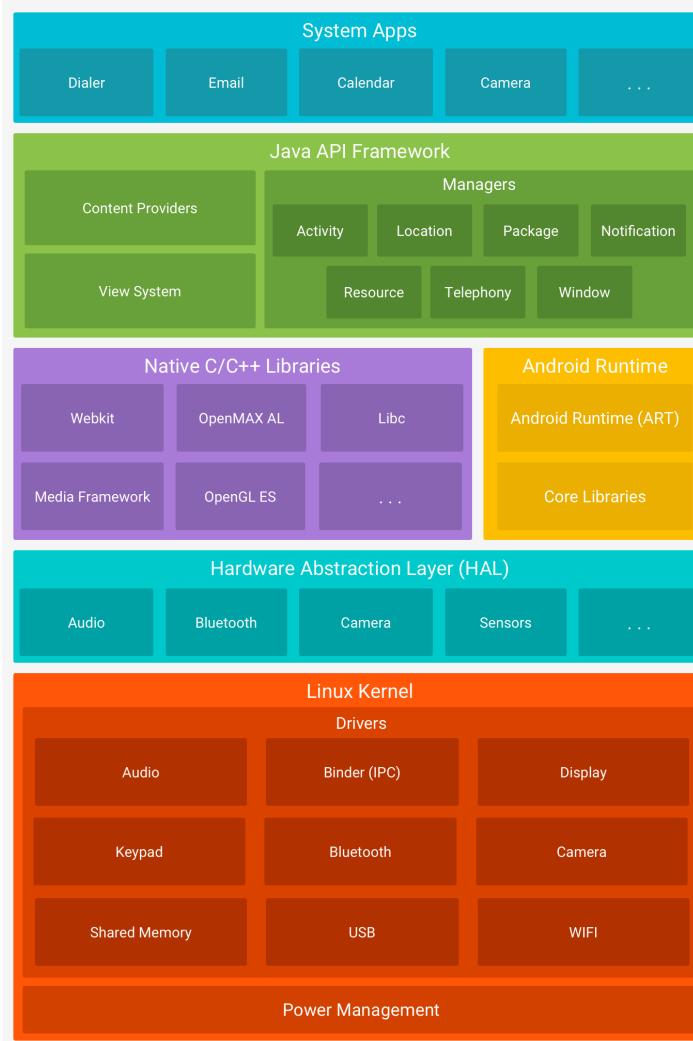
Application framework and libraries in Java

Java libraries based on OpenJDK (Formerly Apache Harmony)

Dalvik as process VM, now Android Runtime (ART) since Lollipop (5.0)



Android Software Stack



Android

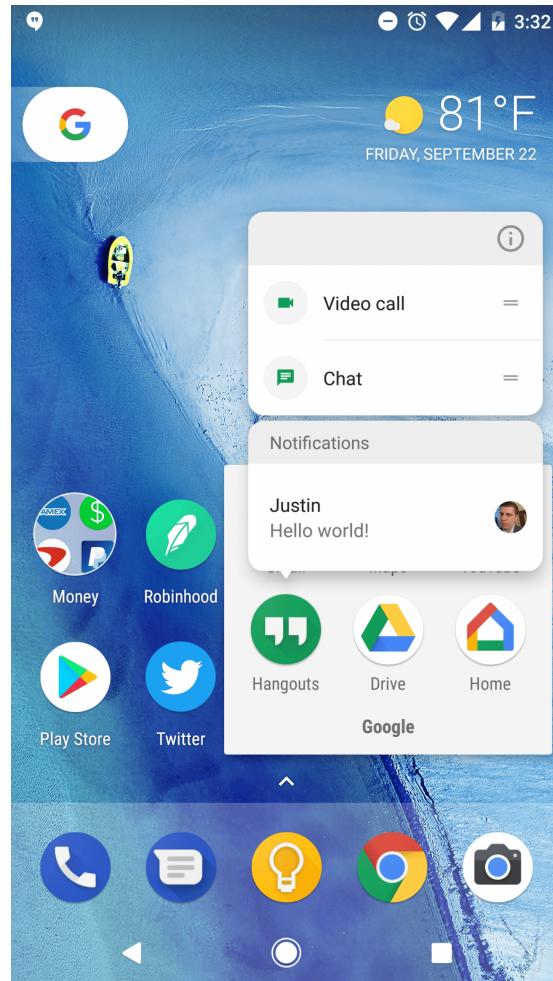
Languages: Java, Kotlin, C/C++

Supports 64 or 32bit ARM architecture, x86 also

Compile to dex bytecode to run on Dalvik or ART virtual machine

IDE: Android Studio

Google Play Store



Other Mobile Operating Systems

Fuchsia (Google)

<https://fuchsia.dev/>

Harmony/HongMeng OS (Huawei)

Mysterious, no information

Tizen (Samsung)

<https://www.tizen.org/>

Summary

Increasing popularity and growth of mobile devices

Mobile device features evolved from single use cases such as calls to multiple features and general-purpose computing

Many different platforms and architectures for mobile devices

Apps can be native, hybrid or web, pros and cons for each