



Mobile Application Development

Introduction To Mobile Application Development

(Beginning Hybrid Mobile Application Development chapter 1)

1-1

Outline

- The evolution of mobile application development
- Different ecosystems: Apple, Google, Microsoft
- Problems with ecosystem-based applications
- Web Sites and Web Views for Mobile Devices
- Adding JavaScript
- Hybrid mobile applications, Front-end and back-end development
- Overview of Testing mobile applications, Native vs. Hybrid Mobile Applications

1-2

History

- Today mobile device users prefer to use applications installed on their smartphones.
- They use installed applications (apps) for carrying out routine activities like booking cabs, buying movie tickets, and watching videos on YouTube.
- The market share of mobile devices is divided mainly into Android, iOS, and Windows Phone.
- Because of the differences in platforms/operating systems, creating an installable mobile application that targets multiple device platforms requires too much of effort and expertise.

1-3

History

- For example, you have to write code in Java for Android, in Objective C for iOS, and in .NET for the Windows Phone.
- Shortcomings of this development approach are as follows:
 - More development time
 - Different expertise required per platform
 - Considerably high cost of development
- This can be overcome by using hybrid mobile applications, a solution based on HTML 5, JQuery, and CSS 3.
- These hybrid applications are created once, but after packaging can be deployed on multiple mobile devices such as Android, iOS, and Windows Phone.

1-4

History

- Key benefits of this development approach are as follows:
 - Less development effort
 - Lower cost
 - Common set of technical expertise required

1-5

History

- 1) Initially, Andy Rubin founded Android Incorporation in Palo Alto, California, United States in October, 2003.
- 2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.
- 3) The key employees of Android Incorporation are Andy Rubin, Rich Miner, Chris White and Nick Sears.
- 4) Originally intended for camera but shifted to smart phones later because of low market for camera only.
- 5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots.

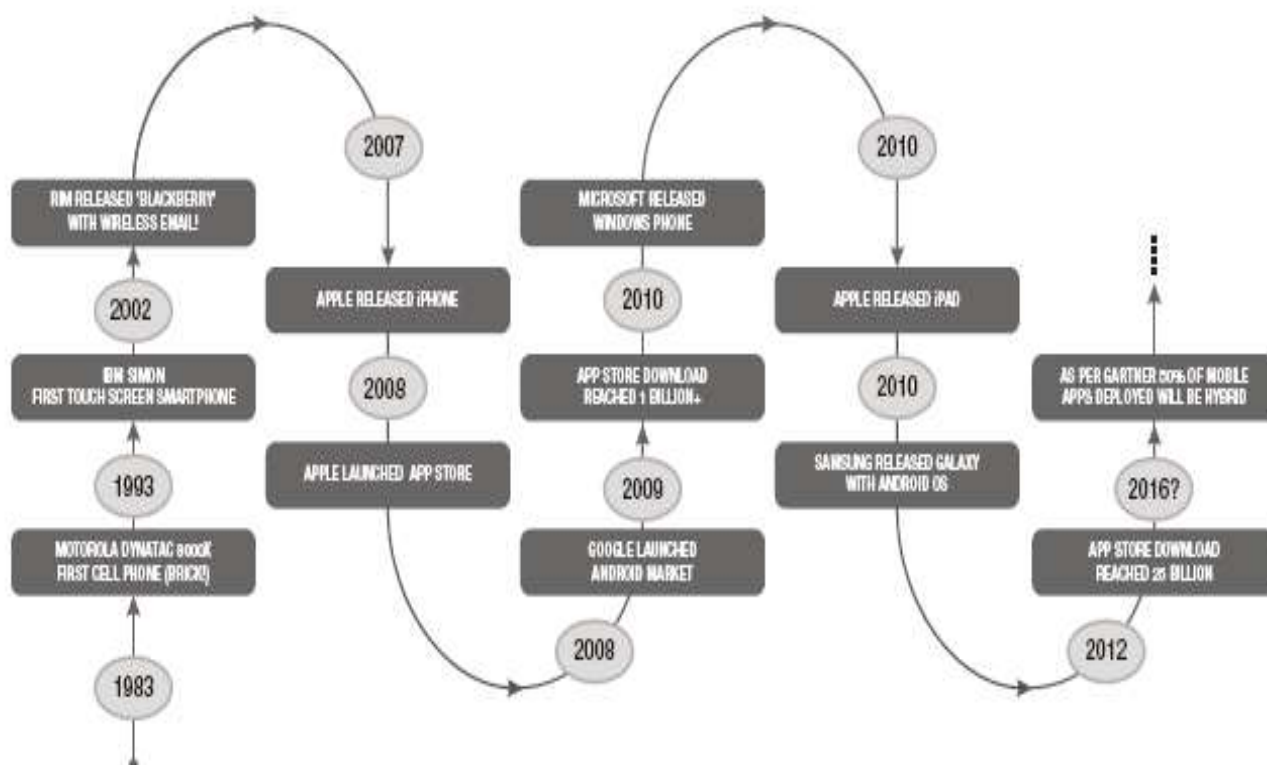
1-6

History

- 6) In 2007, Google announces the development of android OS.
- 7) In 2008, HTC launched the first android mobile.

1-7

The evolution of mobile application development

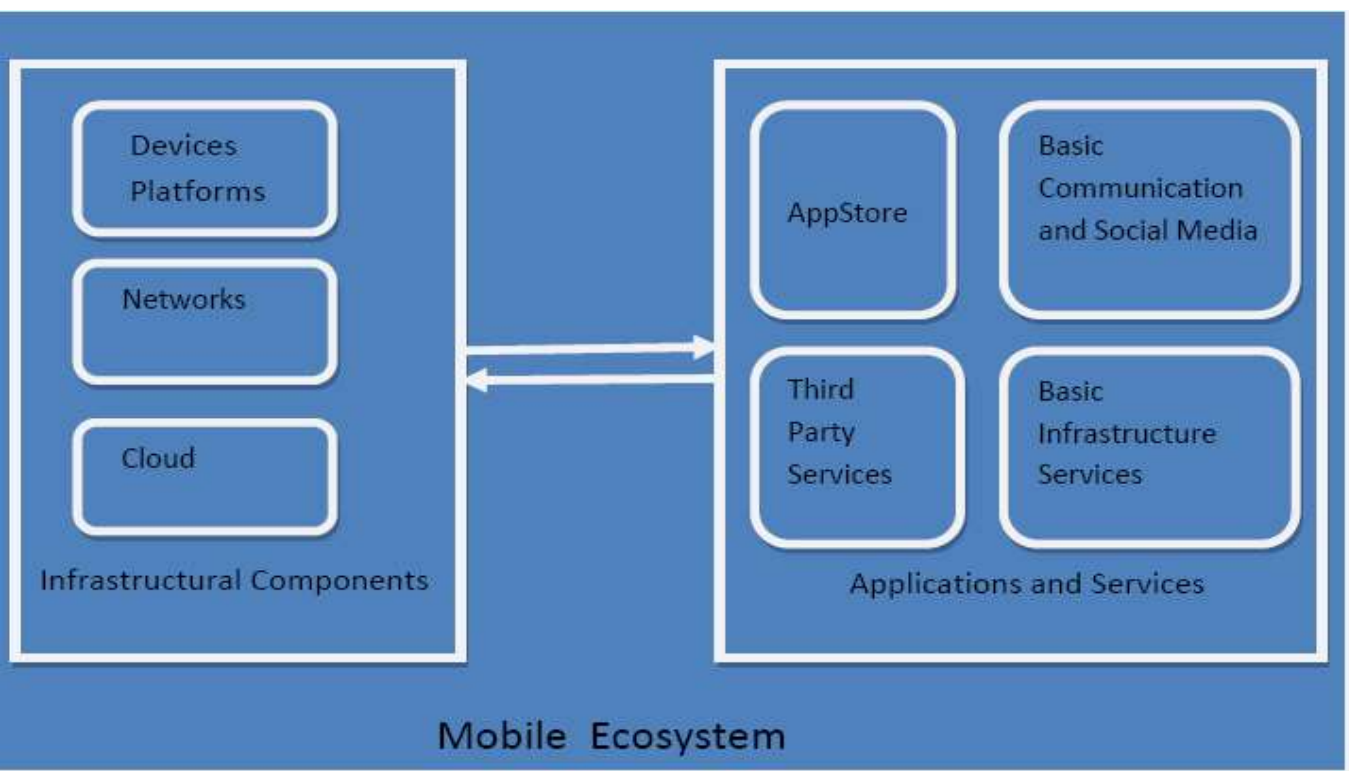


Ecosystems

- How did mobile application development (ecosystems) evolve?
- The three giant vendors/manufacturers in this space: Apple, Google, and Microsoft.
- When a client targets a particular audience for a mobile application, how do you decide which vendor or platform to go with? This depends on multiple factors, ranging from region, location, language, domain, features required, delivery time, development team, and many more.

1-9

General architecture of mobile ecosystem



1-10

The Apple Ecosystems

- Imagine that a client chooses a range of Apple devices for deploying an application.
- The Apple platform is a big ecosystem. As you may know, the iPhone, iPad, and MacBook all fall under the Apple ecosystem.
- Apple also promotes development of applications targeting the Apple ecosystem, by offering many common APIs and a common approval process.
- Development is made easy by Objective C and the Xcode
- IDE, along with many APIs to natively access features in the app such as a camera and location.

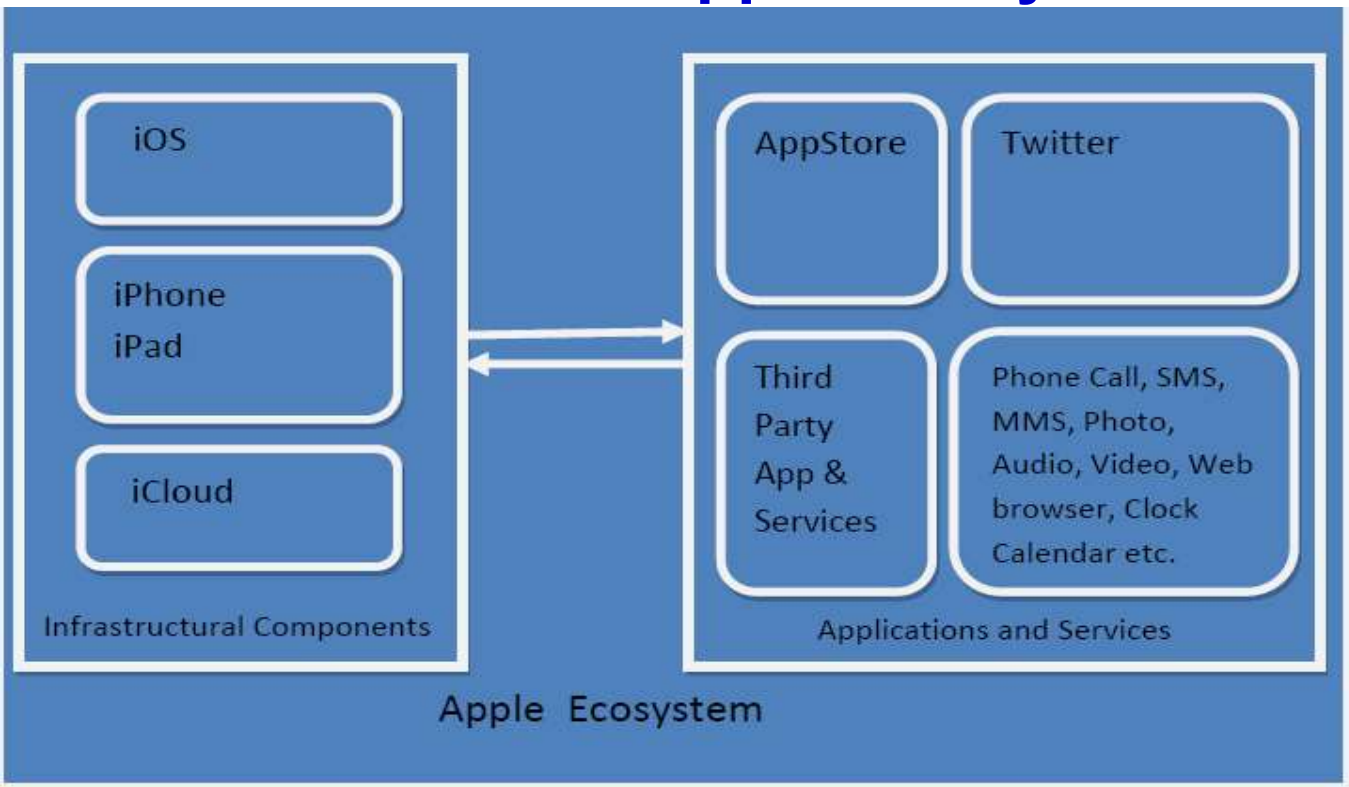
1-11

The Apple Ecosystems

- If the application is a paid one, the revenue-sharing equation is commonly at a 30:70 ratio; Apple takes 30% of the revenue, and the application owner takes 70%.
- Even if the application cost includes charges paid to a service vendor such as SMS gateways, Apple still charges 30% of the total.
- But in any case, the Apple platform is one way for application developers to earn good revenue.

1-12

Architecture of Apple Ecosystem



1-13

The Apple Ecosystems Tools

- Language used Objective C
- IDE Xcode IDE
- Debugger Available with iPhone SDK and installed for Xcode
- Packaging format Packaging through Xcode, distribution is through App Store only. Putting an app on the Google marketplace requires approval from Apple.
- Development tool cost Free on Intel-based Macintosh
- Emulator available Yes

1-14

The Google Ecosystems

- Let's say the same client needs to target the Google ecosystem.
- Do we need changes in the applications?
- Can we port the application as it is?
- Are certification guidelines for Google the same as for Apple?
- Is the revenue sharing equation the same?
- The short answer is no.
- The key thing to understand is that the Google ecosystem is different.
- You will have to consider Android-based devices.
- The Android OS is an open platform to manufacturers, so the market has many device manufactures compared to Apple.

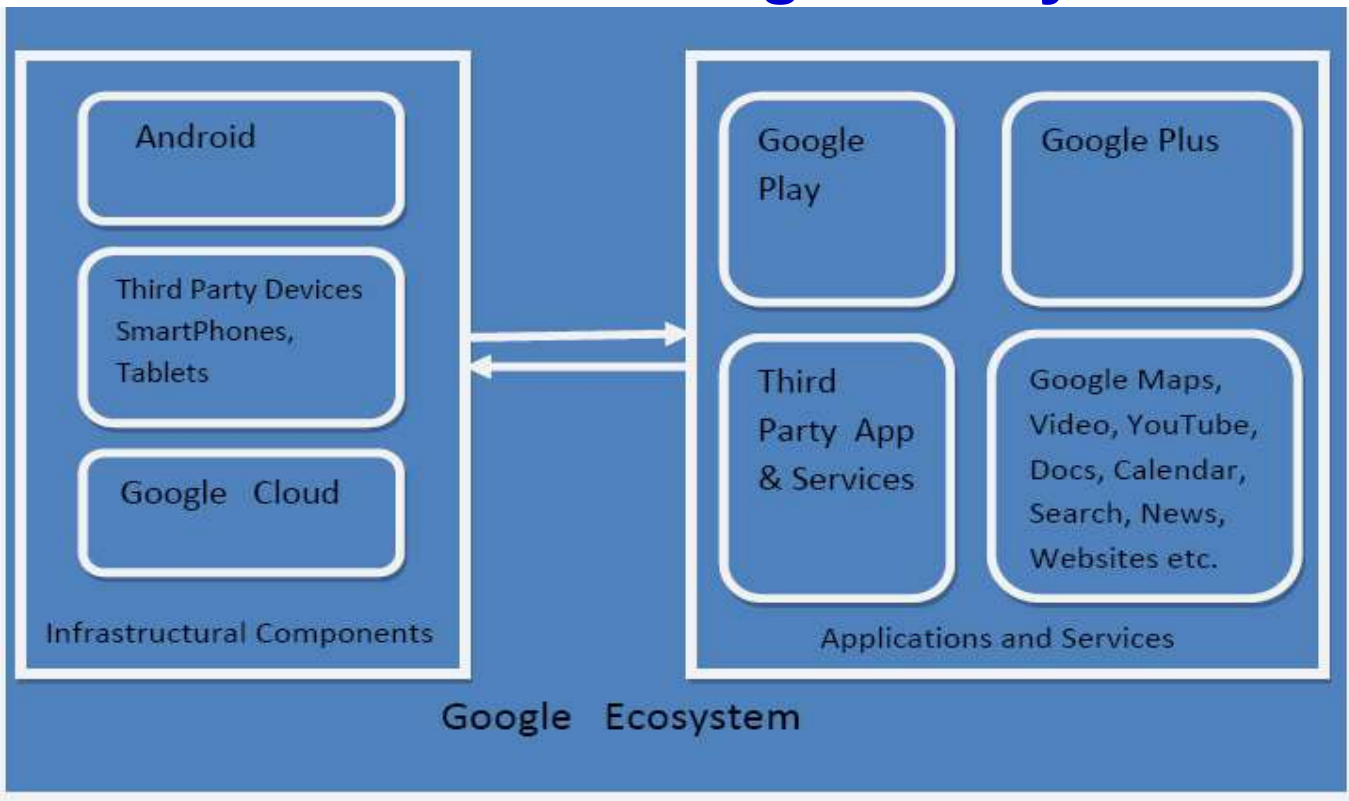
1-15

The Google Ecosystems

- Because Android is allowed to customize, this only boosts the variety of available Android devices.
- Because of the multiple device manufacturers, ultimately many devices vary in size, resolution, and available features, so this ecosystem has many challenges compared to Apple.
- For example, if you develop an application for a Samsung Galaxy Android-based phone, changes may be required if the application is ported to a Google Nexus device, because of resolution differences.
- The good point in favor of Android is that unlike Apple, application development for Android-based devices is mostly in Java.
- This is one of the popular and older languages, so it is easy to find programmers in Java for Android, compared to Objective C or Xcode for Apple.

1-16

Architecture of Google Ecosystem



The Android Ecosystems Tools

- Language used Java. C or C++ can be used for a few component development scenarios.
- IDE Preferred IDE is Android Studio.
- Debugger Android Studio's built-in debugger.
- Packaging format APK. Distribution is through Google Marketplace, as well as directly. Putting an app in the App Store requires approval from Google.
- Development tool cost Free
- Emulator available Yes

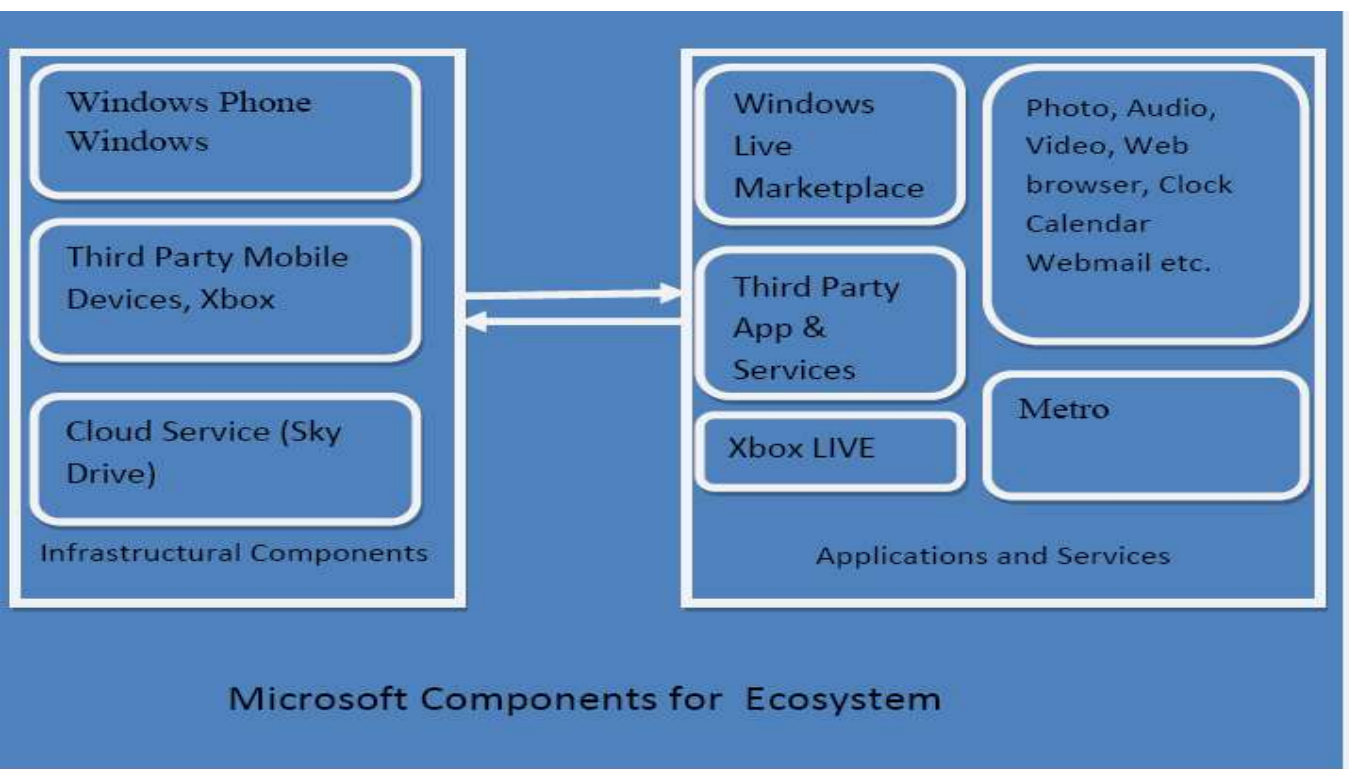
The Microsoft Ecosystems

- The Microsoft ecosystem is similar to that of Apple.
- While working in the Microsoft ecosystem, you have to consider a range of devices, including the Windows desktop, Windows Phone, and Surface.
- Development platforms are the .NET Framework and XNA. Microsoft has the Windows App Store for distributing applications.
- The preferred IDE is Microsoft Visual Studio. Mainly C# and VB.NET are the languages used for development.
- The process and ratio of sharing benefits remains the same as that of Apple (70:30, in favor of the application owner).

1-19

Architecture of Google Ecosystem

Microsoft components for Ecosystem



The Microsoft Ecosystems

- Language used C#, VB.NET, C, C++
- IDE Visual Studio
- Debugger Built-in debugger with Visual Studio
- Packaging format XAP. Distribution through Microsoft marketplace
- Development tool cost Visual Studio 2015 community edition is free
- Emulator available Yes

1-21

Problems with Ecosystems

- As new device are added everyday such as TVs, wearable computers like watch etc.
- So the development is more challenging, to develop for one ecosystem and easily port on another.
- That common ecosystem concept will need to be compromised when developing cross-ecosystem applications.

1-22

Web sites and Web Views for Mobile Device

- Web-view application : An application running under a web browser's context. These kinds of applications seem to be working inside the browser, but without browser windows visible.
- Browsing web sites on mobile-based browsers is common nowadays. Web-site development has made a few terms popular, including front-end and back-end development.
- For web developers, front-end development means designing the user interface (UI) for web sites.
- Backend development means coding behind the UI, and authoring code on the server side by using ASP/JSP/PHP pages.

1-23

Web sites and Web Views for Mobile Device

- When you develop these pages for normal desktop-based browsers, you have the liberty of using huge real estate in terms of screen size.
- When the same pages are viewed or browsed with mobile-based browsers, limitations related to the smaller screen size become the first obstacle.
- How many people really browse web pages on mobile devices? According to Internet.org, “As of October 2014, 55% of cell phone users browse the Web on their devices.”
- How do you tackle the question of screen real estate?
- Responsive design of web pages, which resizes the UI based on the real estate (the screen size), becomes the solution.

1-24

Web sites and Web Views for Mobile Device

- Today, CSS frameworks such as Bootstrap and Foundation provide CSS classes to design and develop responsive web sites.
- Web-site pages are rendered as the UI in a mobile browser.

1-25

Adding JavaScript

- Today, CSS frameworks such as Bootstrap and Foundation provide CSS classes to design and develop responsive web sites.
- The browser, apart from HTML, can also understand one scripting language: JavaScript! , how to deal with native APIs and access device features through JavaScript?
- The web view, along with some code, can then compete with company-specific ecosystems such as Apple, Google, or Microsoft.
- Can HTML-, JavaScript-, and CSS-based installable applications packaged for a particular device or manufacturer replace individual ecosystems? Can we package HTML, JavaScript, and CSS?
- Partially, yes! This type of development is called hybrid application development.

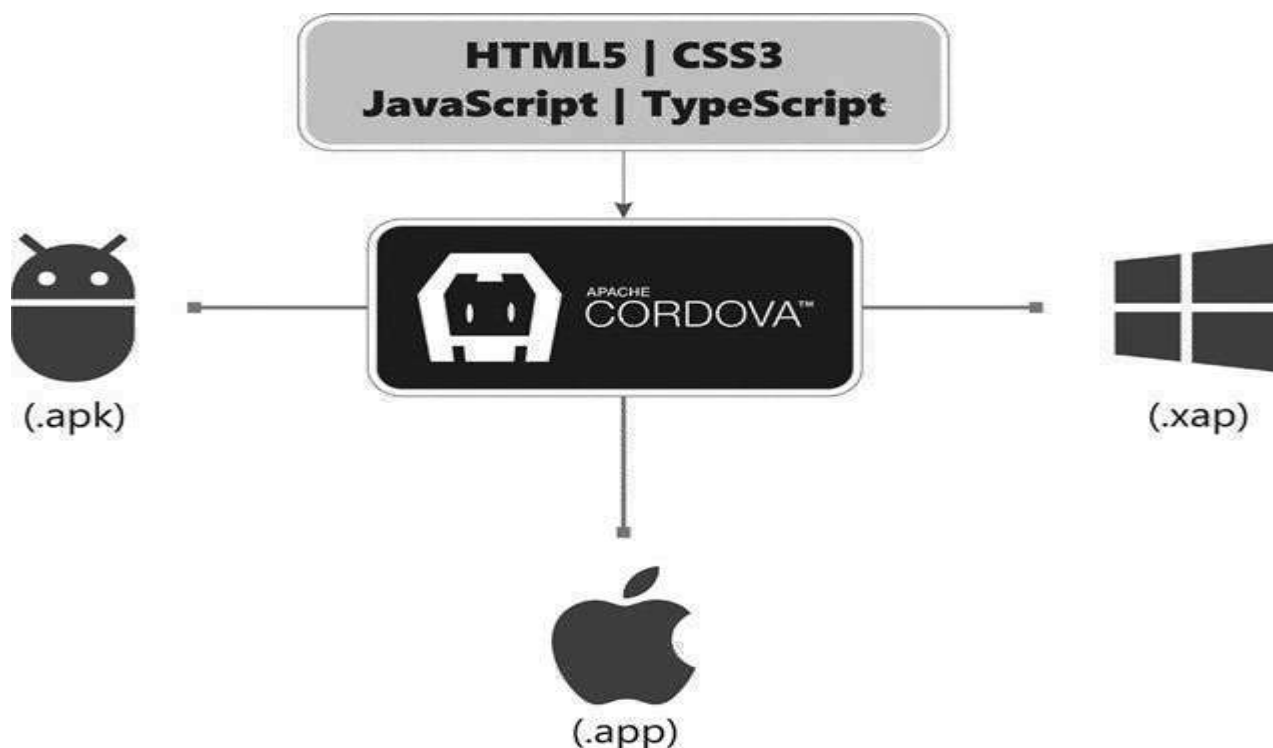
1-26

Hybrid Application Frameworks

- JavaScript executes in the context of the browser, so how can JavaScript talk to devices?
- Hybrid application frameworks such as Icenium, Ionic, and Angular UI can make JavaScript capable of communicating with common devices, in a common way through one common engine or library called Apache Cordova.
- Hybrid and non-native application development in this common ecosystem may have to compromise in terms of the number of APIs available.

1-27

Hybrid Application Frameworks



1-28

Front-End Development

- When considering the front end in web technologies, one term promptly comes to mind: UI design.
- However, the front end is much more than only the UI. You have two separate and confusing terms to understand:
 - User experience (UX)
 - User interface (UI)
- UX is not UI. For example, Your client wants to have a logo—an image—at the top of the home page. Where will you put it?
- UX experts are always confused with UI designers.

1-29

Front-End Development

- UX-ers are better at identifying the following:
 - What customers really want in the interface
 - How to provide a better experience on the small screen
 - Based on the ecosystem, how the application can achieve certification
 - Where to put which component
 - Aesthetic structure and color scheme
- A UI designer might design the interface based on the UX-er's inputs. When considering ecosystem specific mobile applications, such as Apple and Android, the design principles differ.

1-30

Back-End Development

- Back-end development is again a complex term. On the Web, the back end refers to the code that may communicate with a server.
- In a mobile application development scenario, this term can have multiple meanings. Code that you write in a mobile application may do any of the following:
 - Bind data with the UI(manipulate UI components)
 - Get data from the UI
 - Send and receive data to and from server
 - Communicate with APIs offered by the ecosystems to access sensors
 - Execute only business logic without a UI (for example, a service without a UI)

1-31

Mobile Application Testing

- Mobile applications are initially tested with emulators.
- With emulators we cannot test gestures such as a pinch, accessing sensors.
- In this case, we are testing the features of the devices and not the application code testing.

1-32

Android Testing

- To test an app for Android devices, you can copy the program's output files (APK) directly onto the Android device and install it.
- To create a profile and upload applications to Google Play, you need to get a developer license at a cost of about \$25 from Google.

1-33

Apple Testing

- To test a program for Apple devices, you can create a developer profile first at the Apple app store for \$99 a year.
- With the same profile, you can install and test applications on an iPhone.
- Registering 100 devices per product family is a cut-off limit for single developer profile.

1-34

Microsoft Testing

- To test a program for a Windows Phone, you can create a developer profile first at the Windows marketplace, which costs \$19 for an individual, and \$99 for a company.
- With the same profile, you can install and test an application on a Windows phone.

1-35

Hybrid Testing

- For hybrid mobile application testing, you can run the application as a web view inside the browser.
- To test the application's look and feel on different screen sizes, you can use sites such as www.responsinator.com.
- After testing the UI look and feel, you can test the program against sensors or on specific features by installing the application on the device itself.
- But even installing this application on the device requires a license in the Apple and Microsoft ecosystems. Deploying these applications on the stores requires having a profile on the stores.

1-36

Native Mobile Application Development

- In native mobile application development (NMAD), mobile applications are developed using languages supported by the mobile OS technology stack.
- Before Nokia started the mobile application trend, cell phones could do only two things: make calls and send text.
- When a mobile application is created using the mobile OS technology stack exclusively, it is called a native application.
- These applications are built by using only the tools and technologies (including programming languages) suggested by the mobile application stack vendors, such as Google (Android), Apple (iOS), and Microsoft (Windows Phone).

1-37

Native Mobile Application Development

- The advantages are as follows:
 - Better performance
 - Easier development
 - Easy money-making through built-in app store sales
- Here are the disadvantages:
 - Increased development time and costs
 - Content restrictions and guidelines, based on the ecosystem

1-38

Hybrid Mobile Application Development

- In hybrid mobile application development (HMAD), mobile applications are developed using a technology stack and are packaged to deploy on many mobile devices with different screen sizes and manufacturers.
- Hybrid applications allow an application developer to build an application by using simple technologies such as HTML, CSS, and JavaScript. Sometimes developers use C# and VB.NET.
- Hybrid mobile applications try to mix the best of both approaches; they use the power of server-side computing but don't treat the device only as a front end.
- These applications have a native component that resides on the device and can use the local features as if it's a native application.
- That is why hybrid applications are becoming more popular than other approaches.

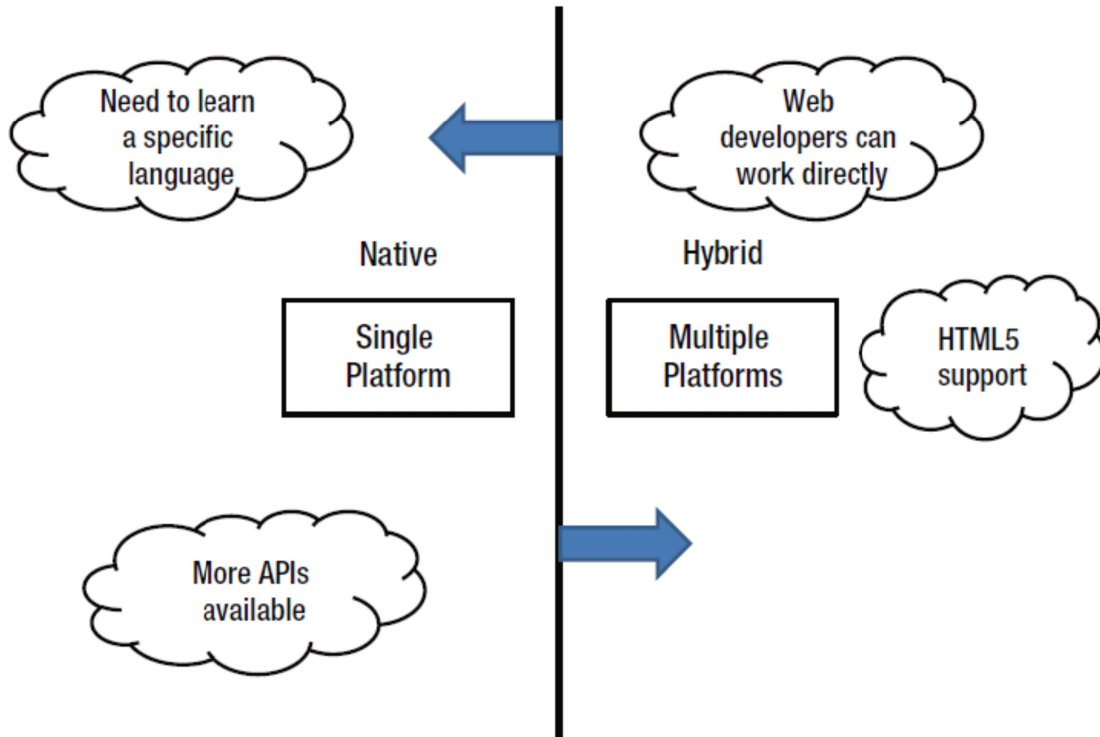
1-39

Hybrid Mobile Application Development

- Here are the advantages:
 - Platform-independent development
 - Easier development
 - Cost-effective

1-40

Hybrid Mobile Application Development



1-41

