Structure of Hybrid Mobile Application Development

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.

Technologies and Frameworks Used in HMAD

- Although, HTML and JavaScript are frequently used for HMAD, the following frameworks are used to communicate with device-based sensors, SD cards, cameras, and so forth:

  - *Ionic: This open source framework focuses on creating good applications in terms of* patterns and practices.
  - *PhoneGAP: More of a packaging tool from Adobe.*
  - *AppBuilder: This is a product from Telerik.*
  - *Kendo UI: Framework from Telerik with the advantage of lots of rich UI widgets.*
  - *Sencha Touch: One of the best framework platforms because of its built-in 50 UI* components.
  - *Angular UI: Mobile Angular UI, similar to Sencha Touch.*
  - *Intel XDK: This framework from Intel comes with an end-to-end development* studio.

Architecture

- We're going to target mobile application development for three ecosystems—Android, Apple, and Microsoft—so first we need to understand the building blocks of those operating systems.

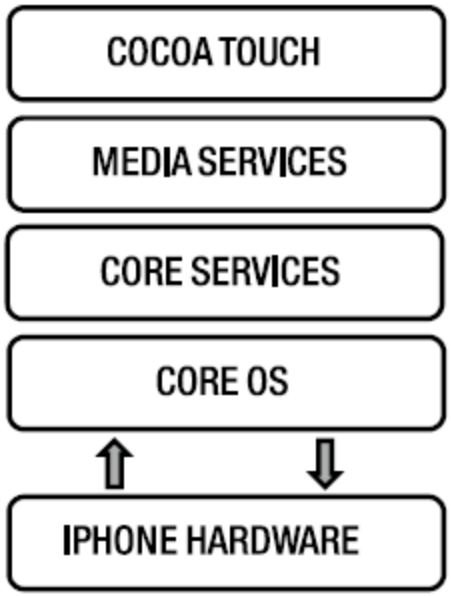
- Each OS has a core component called a *kernel.*

- *How is an API stack built on top of the kernel?*

iOS Layers

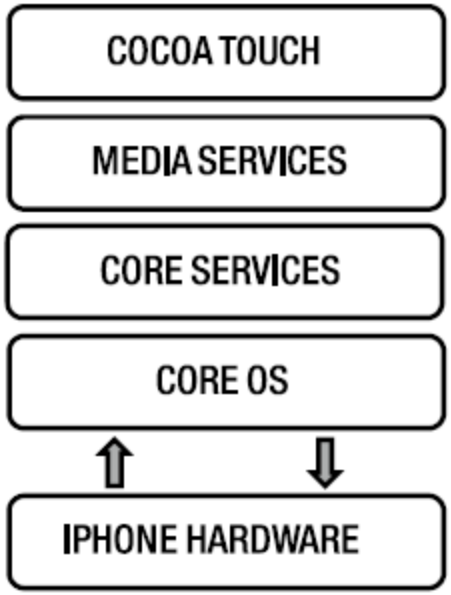
- Each layer in this block diagram has features available, and to access or use those features, many frameworks are also available.

- Code written in Objective C or Swift does not directly talk with the underlying hardware. It has to talk through layers of the core OS in order get access to hardware.



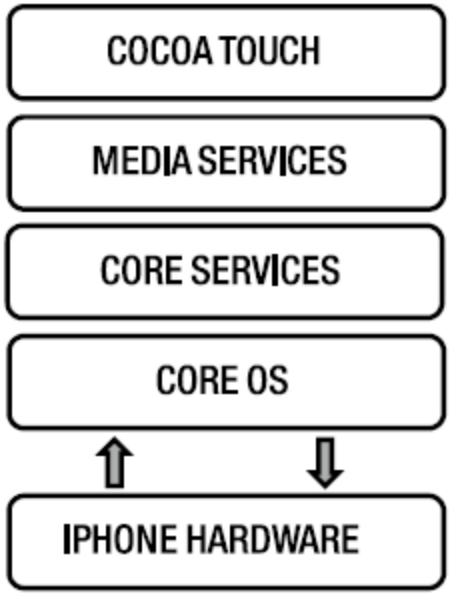
iOS Layers

- Cocoa Touch Layer
  - The first layer is the Cocoa Touch layer. This helps you to define the appearance of the application while porting on the iOSplatform.
  - Various sets of libraries (frameworks) available at this layer help the developer handle different gestures and do multitasking.
  - Here is a list of features in this layer: App extensions, Handoff, Document picker, Air drop, Text kit, UI kit dynamics, Multitasking, Auto layout, Storyboards, UI state preservation, Apple push notification service, Local notifications, Gesture recognizers, Standard system view controllers
  - Useful frameworks (set of libraries) available in this layer are as follows: Address Book UI framework, Event Kit UI framework, Game Kit framework, iAd framework, Map Kit framework, Message UI framework, Notification Center framework, Push Kit framework, Twitter framework.



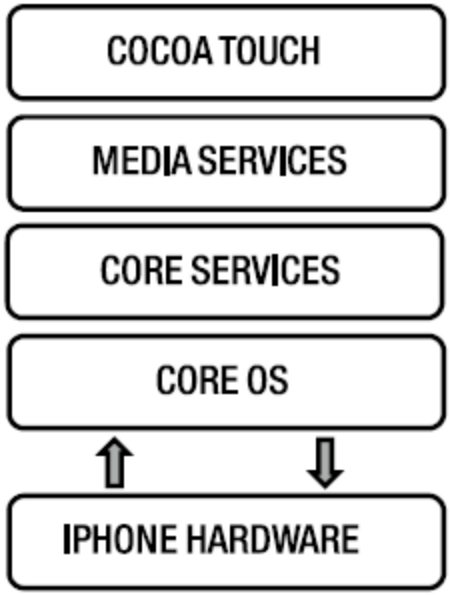
iOS Layers

- Media Services Layer
  - This layer helps you use audio, video, streaming and graphics in the application to implement a multimedia experience.
  - Useful frameworks (set of libraries) available in this layer are as follows: Assets Library framework, AV Foundation framework, AVKit framework, Core Audio framework, CoreAudioKit framework, Core Graphics framework, Core Image Framework, Core Text framework, Core Video framework, Game Controller framework, GLKit framework, Image I/O framework, Media Accessibility framework, Media Player framework, Metal framework, OpenAL framework, OpenGL ES framework, Photos framework, Photos UI framework, Quartz Core framework, SceneKit framework, SpriteKit framework.



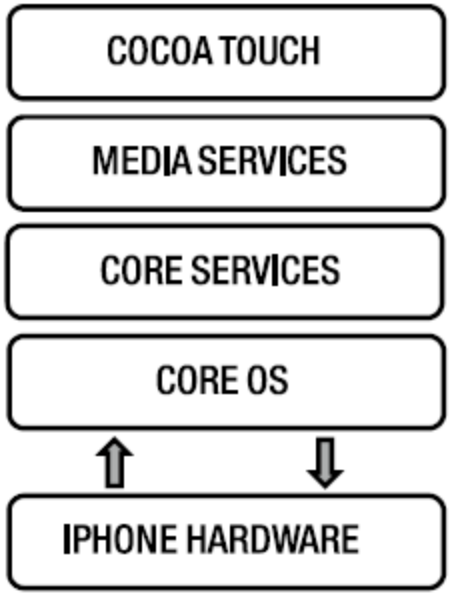
iOS Layers

- Core Services Layer
  - This layer helps you leverage cloud services, social media, and networking.
  - Here is a list of features in this layer: Peer-to-peer services, iCloud storage, Block objects, Data protection, File-sharing support, Grand central dispatch, In-app purchase, Contents, SQLite, XML support, Audio, Video, Graphics, Streaming.
  - Some useful frameworks (set of libraries) available in this layer are as follows: Accounts framework, Address Book framework, Ad Support framework, CFNetwork framework, CloudKit framework, Core Data framework, Core Foundation framework, Core Location framework, Core Media framework, Core Motion framework, Core Telephony framework, EventKit framework, Foundation framework, HealthKit framework, HomeKit framework, JavaScript Core framework, Mobile Core Services framework, Multipeer Connectivity framework, NewsstandKit framework, PassKit framework, Quick Look framework, Safari Services framework, Social framework, StoreKit framework, System Configuration framework,



iOS Layers

- Core OS Layer
  - This layer contains the core APIs to help you deal with the underlying hardware.
  - If you are using frameworks listed in these three layers, they are internally communicating with the underlying hardware through this Core OS layer.
  - Very rarely do you need to use this layer.
  - Here is a list of features in this layer: Security, Access to external hardware, Acting as a bridge between other layers and hardware,
  - Some useful frameworks (set of libraries) available in this layer are as follows: Accelerate framework, Core Bluetooth framework, External Accessory framework, Generic Security Services framework, Local Authentication framework, Network Extension framework, Security framework.



Windows Phone Layers

- Windows Phone architecture has four main layers:
  - Hardware at the base
  - Windows NT kernel on top of it
  - App - UI model
  - Application framework at the top

- Applications for Windows Phone are mainly developed in the Silverlight, XNA, HTML, and JavaScript frameworks.

- The majority of Windows Phone application development occurs with the Silverlight framework. XNA is used mainly for game development. HTML- and JavaScript-based native application development occurs less often than Silverlight.

Browser-Based Applications and Browser Runtime

- To understand the browser runtime from the perspective of hybrid applications, you have to understand browser-based applications.
- We use a browser for requesting pages from web sites and viewing the same.
- Every browser understands web pages, whether static or dynamic, as every page finally gets converted into HTML only.
- The browser is a program that internally has a layered architecture, as shown in Figure

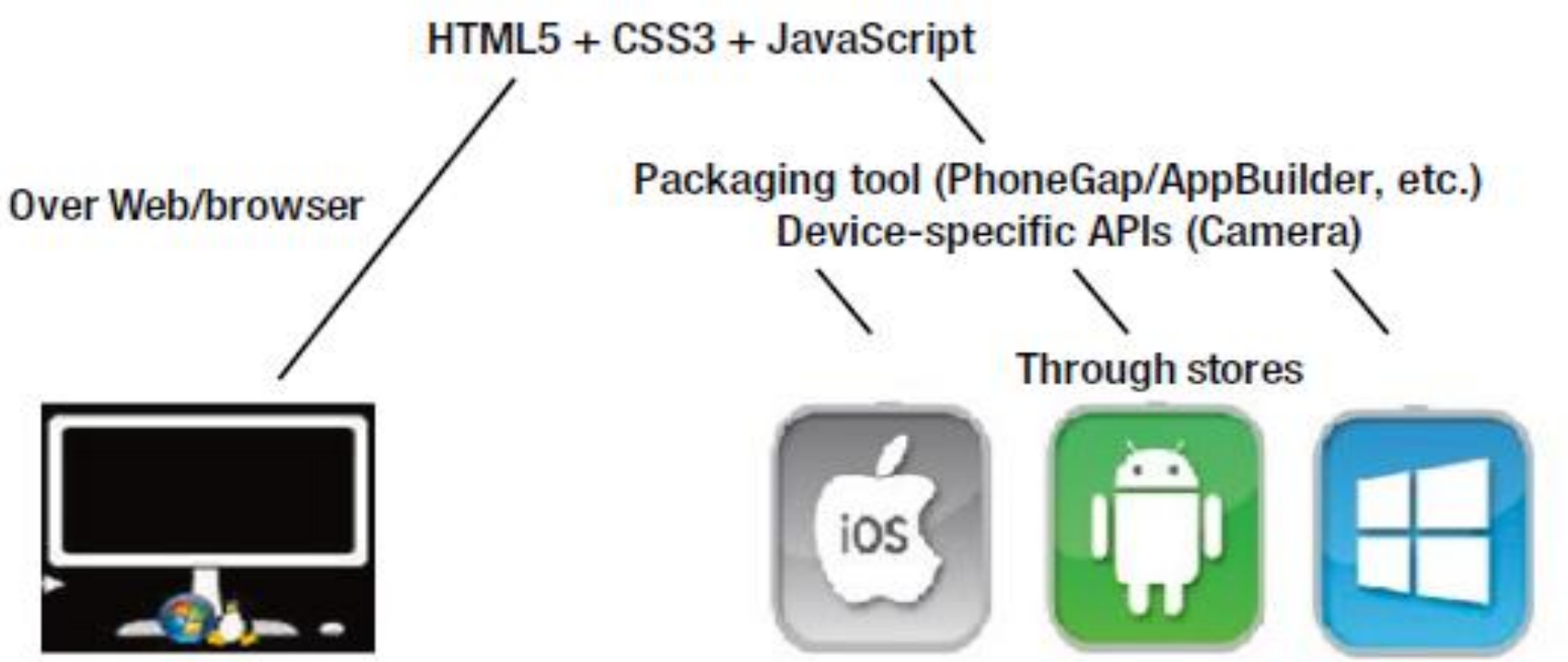
Browser main layers

- All browsers have key components such as the following:
  - User interface : The user interface represents the browser's own look and feel, excluding the area where you can see the requested page
  - Rendering component: The rendering component involves an HTML parser, which, in turn, after parsing can generate the UI as well.
  - Browser engine : The browser engine is the media between the user interface and the rendering engine.
- Each browser, by default, understands HTML and CSS through the rendering engine. It also understands JavaScript code through a JavaScript interpreter.

How Hybrid Applications Work

- If applications for mobile devices can be created with the help of HTML5, what can be used for writing logic (for example, for connectivity to the server and database communication) is JavaScript.

- Figure depicts a hybrid mobile application.



How Hybrid Applications Work

- An HTML5 JavaScript-based web application can be extended to access device-specific features and packaged using frameworks such as PhoneGap, AppBuilder, and Ionic.

- These frameworks also help to package the same app for different ecosystems.

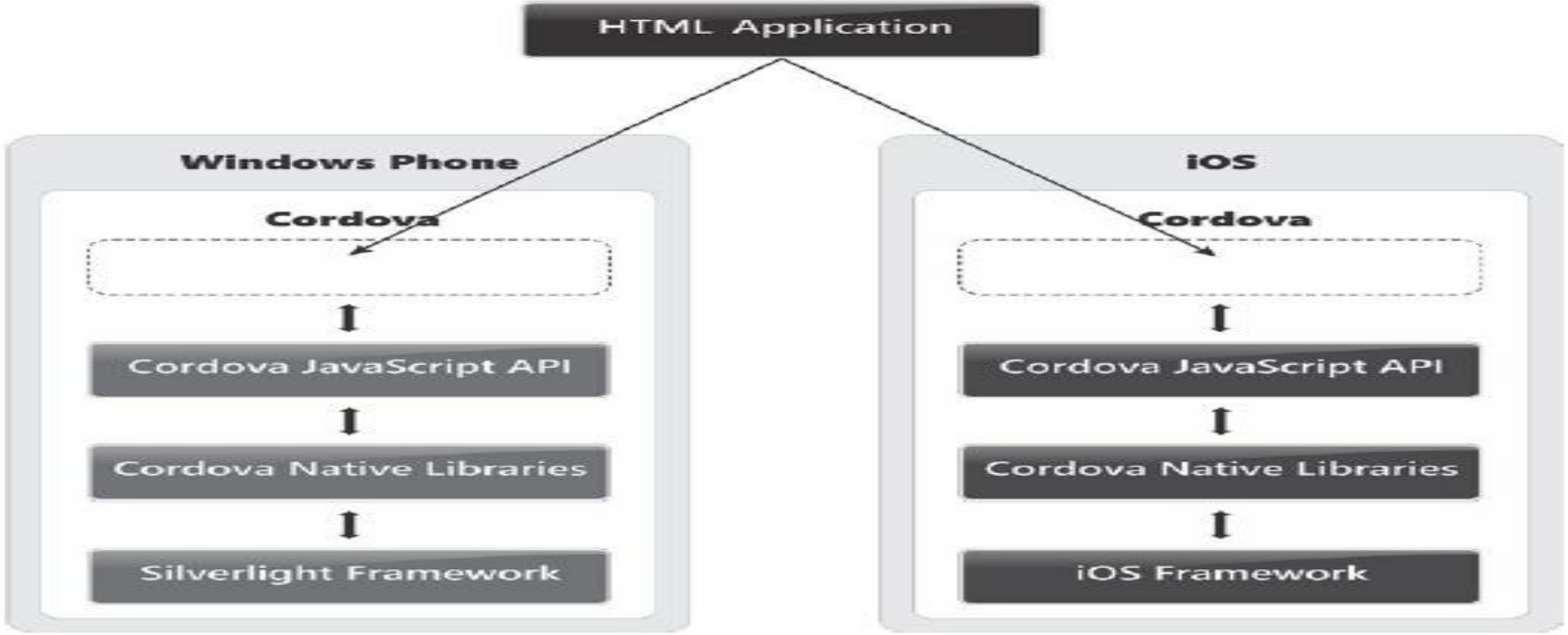
- **Note Most hybrid application frameworks are based on top of the Apache Cordova engine.**

- If you want to write common business logic for web and mobile devices as well, authoring business logic as a service by using service-oriented architecture (SOA) is suggested.

Apache Cordova

- Apache Cordova, from the Apache Software Foundation (ASF), is an API using JavaScript to access device-specific features such as cameras and memory cards. Cordova is open source.[http://cordova.apache.org.]

- These APIs can be combined with frameworks such as jQuery and Dojo. Using this combo, you can develop applications for mobile devices without using languages like Java, Objective C, or C# (see Figure).



How Cordova works

Web Applications vs. Hybrid Mobile Applications

- Because HTML and JavaScript-based hybrid mobile applications use the same UI as that of their web versions, there are chances of confusion between both. Table helps put the differences in black-and-white.

Web Applications with HTML/CSS/JavaScript	Hybrid Mobile Applications with HTML/CSS/JavaScript
Using these applications requires browsing the UI via the browser and raising an HTTP request over the Internet.	Using these applications requires first installing the application on the mobile device. Depending on the nature of the application, you may or may not have to raise the request over the Internet.
JavaScript code written with these applications cannot communicate with the client hardware, as it is restricted to run within the browser premises.	JavaScript code written with these applications can communicate with the devices and the client hardware, with permissions, by using an engine like Apache Cordova.
The user interface can be heavy.	The user interface has to be lightweight, as the UI has less real estate in terms of screen size.

Web Applications vs. Hybrid Mobile Applications

- Both can use the same UI, but that UI has to have responsive design (see Figure).

- Responsive web sites automatically scale the application's UI based on the device's screen size.

HTML5

- HTML5 is a new web markup language standard, announced in 2009 by W3C. The second and final draft, originally scheduled for 2022, was published early in 2014.

- End users are required to install the latest browsers on their machines with the capacity of rendering HTML5 tags and APIs.

- There is no need to write JavaScript code for basic validations, because of the new input types available in HTML5.

- HTML5 supports new input types such as email, color, date, time, and number, which do validation as well (see Figure)

Basics of HTML5 and Useful APIs

- To convey to the browser that the page is written with the HTML5 standard, the first tag is important. The line is as follows:
  <!DOCTYPE html>

- Following are the new structural tags in HTML5:
  - section presents generic content in group format.
  - article presents a self-contained composition.
  - aside presents complete tangent contents (for example, advertisements) compared to the rest of the page.
  - header represents a group of navigational aids.
  - footer presents normal footer content.

- The following are the new content tags in HTML5:
  - figure presents flow content (which may be an image), with a caption preferred.
  - video shows video data; supported formats differ, based on the browser.
  - audio presents audio data; supported formats differ, based on the browser.
  - embed is used for hosting an external application.
  - canvas is used for rendering graphics.

Basics of HTML5 and Useful APIs

- The following are the new application-focused tags in HTML5:
  - meter presents a range or scale of values.
  - progress presents the amount of completion of a task.
  - time presents a 24-hour clock.
  - details is used for conveying additional information.

- The following are deprecated in HTML5 due to replacement through CSS:
  - big
  - center
  - font
  - strike
  - U

- The following are deprecated due to accessibility:
  - frame
  - frameset

HTML5 Detection

- You can detect whether HTML5 is supported on a client-side browser in multiple ways; for example, you can use TAG with inline content. If the tag gets rendered, HTML5 is supported. If the content gets rendered, HTML5 is not supported.

```
<canvas style="height:100px, width:100px, background-color=yellow">
        HTML 5 is not supported
<canvas>
```

- Using HTML5 specific APIs, you can check whether the API works; if it doesn't work, you'll see the respective message:

```
if(typeof(WebWorker) == 'object')
{
        alert ('HTML 5 supported')
}
else
{
        alert ('HTML 5 not supported')
}
```

- You also can use the Modernizr library to check whether HTML5 is supported.
- Modernizr is a JavaScript library that can be downloaded from https://modernizr.com/.

HTML5 Specific APIs

- In HTML5, APIs can be divided into two categories: integrated and associated.

- **Integrated APIs** : Integrated APIs may not require any script code. These can be directly used, and be presented through TAG or file. Such APIs include Video, Audio, and Drag and Drop.

    - **Video** : The video tag does not require a third-party plug-in installation such as Flash:

```
<video src="<VIDEOFILE.EXTENSION>" width="100px" height= "100px"
><.video>
```

    - In the preceding example, VIDEOFILE.EXTENSION represents the pattern used to provide the source.
    - The following codec formats are recommended by vendors:
    - H.264 by Microsoft and Apple (conversion through the HandBrake utility)
    - OGG by Firefox and Opera (conversion through Firefogg on Firefox)
    - VP8 by Google and Mozilla (conversion through the HandBrake utility)

HTML5 Specific APIs

- **Audio** : The Audio API is used for playing audio content over the browser, without a plug-in requirement.
- Supported formats include MP3 and WAV.
- This API can help the user browse content offered from the server side offline. You can convey to the browser what kind of resources need to be in cache at the client side.
- The file used for specifying the cache part is standard, and is named cache.manifest. This file is required to be provided with the HTML tag itself, as follows:

<html manifest="cache.manifest">
The cache.manifest file is shown in Figure

```
CACHE MANIFEST
# this is a comment
CACHE:
index.htm
_css/main.css
_scripts/contacts.js
NETWORK:
updateContacts.cgi
```

HTML5 Specific APIs

- **Drag and Drop** : This API enables you to drag one UI element and drop it at a particular container location.
- A few events (such as onDragEnter, onDragStart, and onDrop) need to be handled to achieve this action.

- **Associated APIs** : It require writing of the script code. Such APIs include Geolocation, 2D Canvas, Local Storage, Web Worker, and Web Sockets.
  - **Geolocation** : This API helps get the end user's current location.
  - The API is added as an extension to the navigator object and provides methods such as getCurrentPosition() and watchPosition().
  - To obtain the position of the end user, permission is required.
  - The reply from the API consists of latitude and longitude, which later can be integrated with the Google Maps or Bing Maps APIs.

  - **2D Canvas** : The Canvas API can be used to draw 2D graphics. Geometry elements such as a circle, rectangle, and line can be drawn with the help of the Canvas API. Canvas becomes a base for graph APIs in many application frameworks.You have to use a 2D context to get the x and y axes before drawing. There is no 3D context available as of now..

HTML5 Specific APIs

- **Local Storage** : In normal cases, browsers support cookies. At a first visit, web servers and web sites may persist a cookie on the client machine to mark that the user has visited the server at that particular time. It may additionally store the user's preferences.
- Upon the site being requested again, the cookie information can be packaged by the browser and sent to the web server or web site, which then can be used to identify the client or user's preferences.
- Although cookies are stored on the client's machine, they are always used by the server when sent back by the browser.
- HTML5 offers a Local Storage API by extending the traditional window object. You can use this API to store about 5MB of data in the key/value pair format on the client's machine, based on the site's domain.

HTML5 Specific APIs

- **Web Worker** : Client-side JavaScript is always a single-threaded environment; multiple scripts can't run at the same time.
- The Web Worker API in HTML5 brings threading to JavaScript. Now, you can run multiple scripts on the client side by using a web worker.
- You have to create a Web Worker object and assign a script to be executed as input in the form of a JS extension form.

- **Web Sockets** : Over the Web, we use HTTP to communicate between a client browser and web server. HTTP is a stateless protocol; no continuous connection exists between the client and server.
- How will the server remember the client in the next trip? We use cookies and session concepts. Using these concepts, you can achieve stateful behavior.
- The Web Socket API in HTML5 offers full, two-way (duplex) communication between the client and server. Underneath, it uses TCP.
- This becomes the base for web applications, which use video streaming or critical (or real-time), large data updates in the UI.

Data Formats

- Whether you use HTML5 for web or hybrid mobile applications, communicating with the server is required for enterprise applications.
- For communicate with server two industry standards available: XML and JSON.
- **Using XML** : This standard became popular as many ecosystems launched APIs to support XML, including Microsoft, Apple, and Google.
- After XML, browsers were launched along with an XML parser. Every browser supports an API for HTML as well as XML, called the Document Object Model (DOM), along with object support including Document, Window, and Navigator.
- When JavaScript, as a client-side coding language, and DOM did not support XML natively, support was added externally by browser vendors including Microsoft, through the MSXML API.

Data Formats

- Logic hosted on the web server and offered with the endpoints exposed is normally known as web services. Web services became the base for distributed enterprise applications.

- Another thing that made XML more popular was web services.

- Web services used XML over HTTP for communication, and were used as a middleware for cross application platform communications such as Java and .NET.

- **Drawbacks of XML** : Though enterprises continue to use XML, it comes with one major drawback. XML data is always written in nodes, to those for whom only data matters, it seems to be carrying redundant parts.

Data Formats

- Consider XML presenting an employee's data:

```
<?xml version="1.0" encoding="UTF-8"?>
<Employee>
        <Name> John </Name>
        <Id> 1928 </Id>
        <Age> 34 </Age>
        <Address>
                <City> Sydney </City>
                <Country> Australia </Country>
        </Address>
</Employee>
```

- For someone who wants to know the data, knowing only that the John value stands for Name is sufficient.

- However, the XML syntax uses the end tag </Name>. Carrying this end tag as a set of characters makes the XML bulky. Using JSON format helps reduce the amount of data transfer.

Using JSON

- JSON stands for *JavaScript Object Notation The preceding XML about the employee can be presented in* JSON as follows:

{ Name: 'John' , Id: '1928' , Age: 34 , Address: { City: 'Sydney' , Country: 'Australia' } }

- You have no end tag as in XML, JSON format is shorter. Even complex data such as an address can be presented.

- Earlier there was no server-side support from popular frameworks like the Java server-side APIs and Microsoft .NET.

- Over the Web, when a response is given by the web server, it always adds a content type. The default content type for response pages has always been text or html. As for XML, it was application or XML.

- When JSON data used to arrive in client-side JavaScript, it was always taken as a string. Programmers used to get actual data by doing a string-split operation, until a helper file (json.js) was offered by Crockford.

Who Uses JSON?

- Today web giants including Twitter, Microsoft, Facebook, and Google use JSON format for communication from server to client, and vice versa.

jQuery

- jQuery is a library based on JavaScript.
- Its code helps solve a major problem for a client script developer (the cross-browser issue).
- The jQuery library comes in two flavors: a development version and a production version.
- jQuery provides several APIs to manipulate the DOM and handle events in a cross-browser way.
- It also heavily supports the JSON format. jQuery AJAX is comparatively easier than normal AJAX, with plain JavaScript.

jQuery Basics

- Because jQuery is ultimately a library in JavaScript itself, if the client browser has restricted execution of JavaScript on the machine, even jQuery is restricted.

- Like JavaScript, a jQuery library needs to be included in the script tag.

- If you have to ask jQuery to search for an input text box with an ID such as txtName, then if you write the code as follows, the code simply fails.

- To use commands in jQuery, you have to use shorthand such as jquery or $.

- Of the two, use of $ is more commonly seen. Here, putting # before txtName means “find by ID.”

jQuery Basics

```
-------------------------------------------------------------------------------
//.....................HTML ..............................
//.....................HTML ..............................
//.....................HTML ..............................

<script language="text/javascript">

var resultName = jquery('#txtName').value(),                    //CODE FAILS HERE
alert(resultName),
</script>
//.....................HTML ..............................
//.....................<txtName element> ......
//.....................HTML ..............................
-------------------------------------------------------------------------------
```

- You would get an error conveying *undefined element 'txtName'! (Undefined what?)*

- The problem is that you're asking jQuery to execute the code way before the txtName control is loaded into the DOM hierarchy. So it cannot find it, because it tries before txtName is even created!

jQuery Basics

- To ensure that the code gets called after the DOM tree and elements are loaded, you need to use the ready() function in jQuery as follows (replacing the preceding script block):

```
<script language="text/javascript">
        $(document).ready(SayHi), //this will execute code in 'SayHi' function
        function SayHi()
        {
                var resultName = jquery('#txtName').value(),
                alert(resultName),
        }
</script>
```

- jQuery prefers inline functions without names, which makes the jQuery syntax unique.

jQuery Selectors

- One of the unique selling points for jQuery are selectors, which allow page elements to be selected.

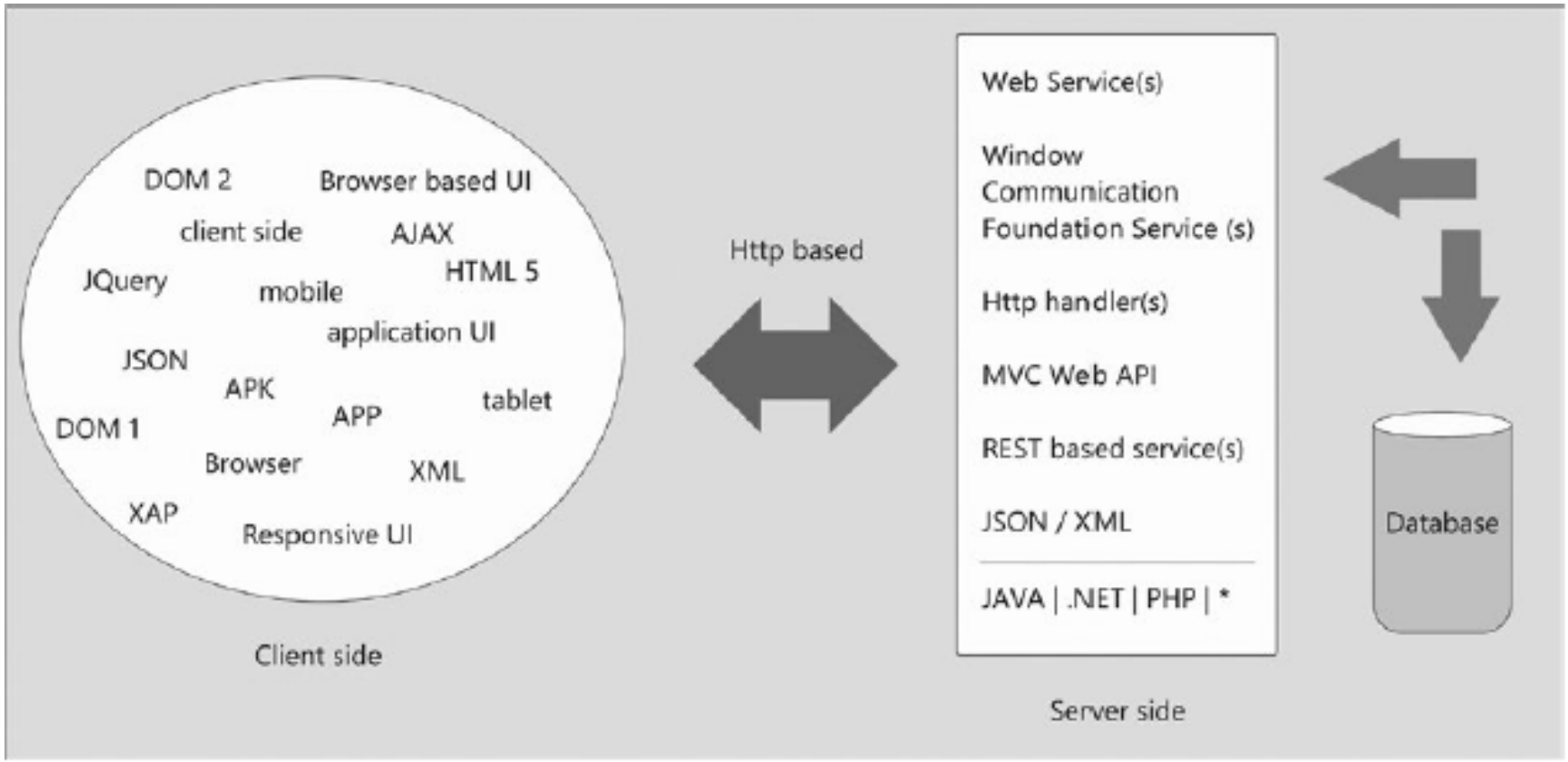
- Selector syntax is as follows:
  $(selectorExpression) or JQuery(selectorExpression)

- The following basic selectors exist in jQuery:
- To select nodes by tag name, use the tag name:
  $('div') selects all div elements
- To select nodes by ID, use #:
  $('#mydiv') selects elements with the ID attribute set to mydiv
- To select nodes by style class name, use .:
  $('.myStyle') selects all elements with CSS style class with the name myStyle applied
- To select nodes by attribute value, use square brackets:
  $('input[type="text"]') selects all input elements with type = "text“
- To select nodes by input nodes:
  $(':input') selects all input elements irrespective all their type

jQuery APIs

Function Name	Description
`.val()`	Returns the content assigned to the value attribute of a tag; for example, `input`.
`.attr('attribute', 'val')`	Works as a getter/setter for a particular attribute of an HTML tag.
`.css(json)`	Sets the values of attributes of a style property. The bulk are set in JSON format.
`.html()`	Gets the inner HTML value of container tags, such as `div`.
`.filter(selector expression)`	Works as the next-level filter to get a specific element in HTML.
`.toggle()`	Toggles between multiple classes applied on the HTML element.
`.append() , .prepend()`	Appends or prepends the content to a container. The selector is a container.

Server Side Support



- **HTTP Handlers** : Techniques for consuming input data, executing business logic, and returning the respective data.

Service-Oriented Architecture

- Service-oriented architecture (SOA) is “a set of components that can be invoked, and whose interface descriptions can be published and discovered.”

- These components can be created and exposed with the help of frameworks such as Java server-side APIs, .NET, and PHP.

- **Web Services** : Web services are web applications that do not produce HTML. They contain logic that accepts input in the form of XML over HTTP and reply in the same manner.

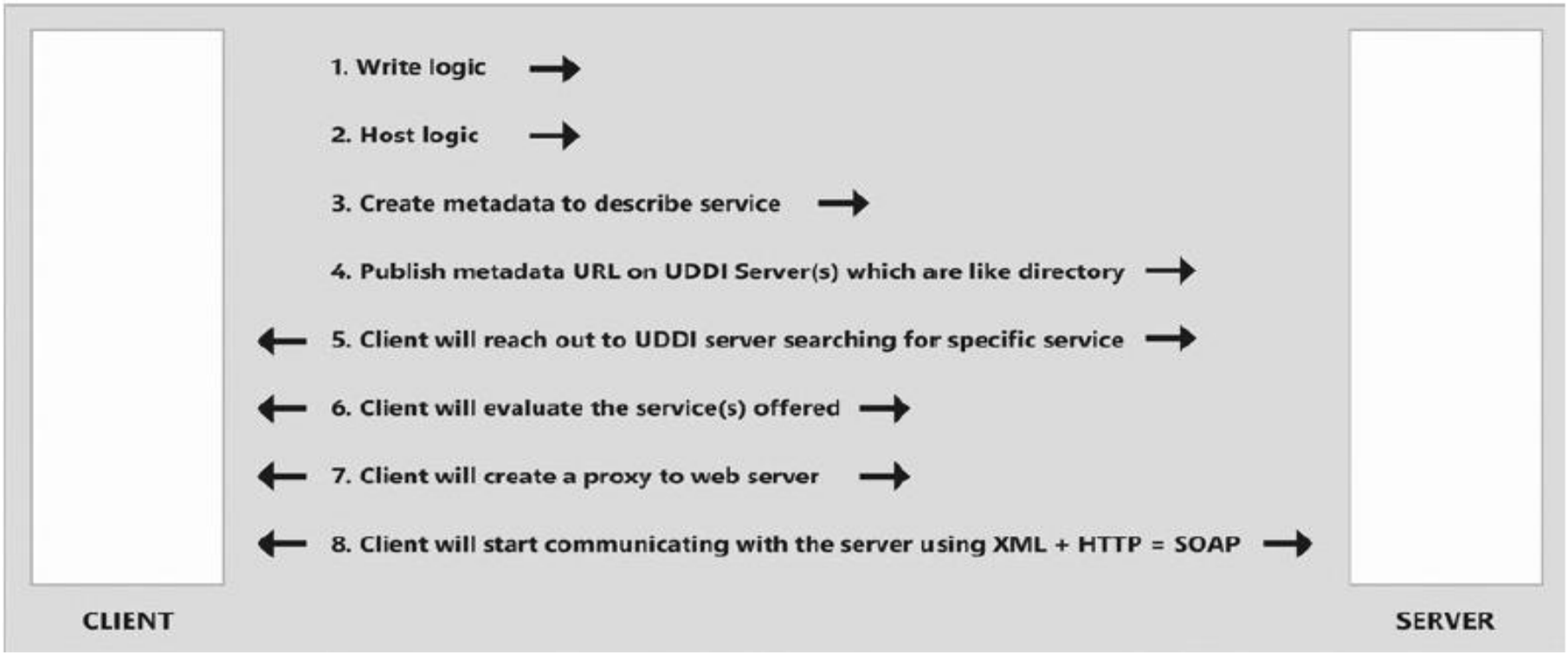
- Based on who is going to call the logic or share the logic, and whether there are multiple clients, you can choose an SOA.

- When you pick the web services approach, you first need to create a business component, which will be shared.

- This component can be called over HTTP. Data transfer happens strictly in XML.

Service-Oriented Architecture

- XML is transferred over an HTTP carrier.
- Technically, XML over HTTP along with a schema definition is referred to as SOAP (originally an acronym for Simple Object Access Protocol) .
- Figure shows the steps taken. The arrows in the figure convey whether the step is taken in the server or client context (or both).



Windows Communication Foundation Service

- Windows Communication Foundation (WCF) services have the following features:
  - Service oriented, just like web services
  - Multiple protocol support such as HTTP, TCP, .NET MSMQ, and P2P
  - Multiple data formats are supported
  - Built-in security APIs
  - Reliable messaging using MSMQ (Message Queuing)
  - DB transaction participation support

REST-Based Services

- REST means *Representational State Transfer.*
- The main idea behind REST is that you should treat services as a resource., which could be access by HTTP.
- WCF with REST is an easy combination.
- REST offers end users the choice of XML or JSON.
- Users can be exposed to all the CRUD (create, read, update, and delete) operations using WCF REST by decorating existing logic classes with attributes.

Cascading Style Sheets (CSS) version 3

- You can use CSS to describe how an HTML element should look.

- Launching CSS3 along with HTML5, W3C offered the following additions to the existing CSS functionality:
  - Selectors, as in jQuery
  - New backgrounds and borders
  - New text effects
  - New 2D/3D transformations
  - Rounded corners
  - Multiple column layout

- **Responsive CSS** : This feature of CSS will make the same web UI appear differently, depending on the user's device size.

- If the application or web site is opened on a mobile device, tablet, or desktop browser, the UI will present as per the real estate in terms of screen size available.

Twitter Bootstrap

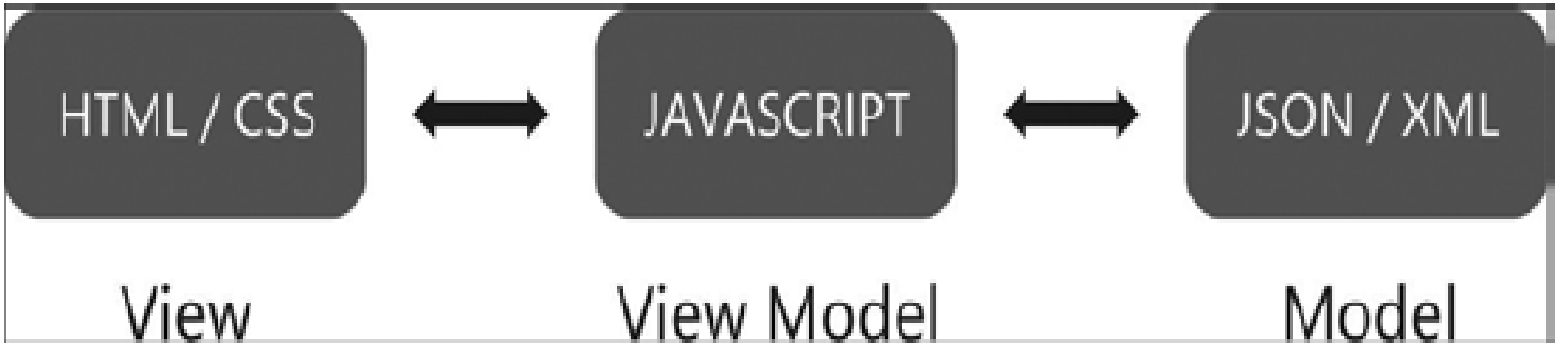
- Twitter Bootstrap provides responsive CSS along with a 12-column grid layout.
- Instead of specifying the size of the control, you can assign the class name to the HTML element offered by Bootstrap CSS. These class names help the element occupy a length as per 12-column grid.
- The available screen size is normally divided into 12 columns. Based on the size requirement, each element is offered 1–12 column-level classes.
- **Skeleton** : Skeleton is also a responsive CSS library with a 12-column grid layout available.
- The Skeleton library is simpler to use than Bootstrap.
- There are fewer UI tools available that work with Skeleton compared to Bootstrap.

HMAD Development and Packaging Frameworks

- What is a packaging framework? While creating HTML5 and JavaScript-based hybrid mobile applications, you can choose the base and packaging from a platter of frameworks.
- Packaging frameworks help you create a final deployable output based on the ecosystem.
- The frameworks helps to port the same code on multiple ecosystems.
- All these frameworks enable applications to compile on top of the Apache Cordova engine.
- Applications built on platforms are always going to run in web view.

Inoic

- This open source framework focuses on creating good applications in terms of patterns and practices.
- Ionic uses the AngularJS library for code purposes.
- It also recommends code in a Model-View-View Model (MVVM) design format, while presenting the browser (client) side alone.
- Three layers exist in the MVVM design pattern, as shown in Figure:



How MVC layers talk with each other

Inoic

- *Model:* JSON or XML data objects that may be received from the server
- View: HTML UI or CSS
- *View Model:* JavaScript code that can bind *JSON or XML* data (the model) with the UI (the view)
- Applications created with Ionic include the following:
  - *ChefSteps:* Cooking application
  - *Mallzee:* Personal styling application
  - *Sworkit:* Body and fitness application

PhoneGap

- PhoneGap is more of packaging framework.
- You can choose a JavaScript library for code like jQuery, plain JS, and AngularJS.
- After coding is done, PhoneGap can package the same, based on the need.
- PhoneGap is owned by Adobe and is based on top of the Apache Cordova engine.
- PhoneGap supports compilation in the cloud (using Software as a Service) without installing the SDK required for packaging applications based on ecosystems.
- Some of the applications created with PhoneGap are as follows:
  - *snowbuddy:* More of personal assistant or buddy type application
  - *indoona:* Chatting application
  - *jigsaw:* A jigsaw game application

AppBuilder

- This product from Telerik was formerly called Icenium.

- AppBuilder has many similarities with PhoneGap, including having Apache Cordova at its base.

- AppBuilder extends compilation-in-the-cloud functionality (just like PhoneGap) to offer code with tools such as these:
    - Command-line interface (CLI)
    - In-browser client
    - Windows-based desktop application

- It also offers a plug-in to popular IDEs such as Visual Studio and Sublime Text.

- The following are some of the applications created with AppBuilder:
    - *Survey Data Plus Plus:* Survey automation software
    - *Birthplace:* Healthcare application

Kendo UI

- Kendo UI is another professional framework from Telerik, but with the advantage of lots of rich UI widgets.

- Kendo UI has more than 70 UI widgets and more than a dozen built-in UI themes.

- It supports AngularJS and Bootstrap.

- Some of the applications created with Kendo UI are as follows:
  - *Udemy:* Education-related application
  - *LunchBoat:* Social meal planning application

Angular UI

- Mobile Angular UI is just like Sencha Touch.
- An important fact to note is that it has tailor-made Bootstrap.
- Bootstrap does not directly offer sidebars or bottom navigation bars.
- Mobile Angular UI helps you get all of that along with Bootstrap!
- Mobile Angular UI is free and open source.

Sencha Touch

- One of the best framework platforms for building hybrid applications is Sencha Touch.
- It is a paid framework, which comes with more than 50 UI components often required in enterprise applications.
- It is fast in execution.
- It uses a Model-View-Controller (MVC) design pattern, considering code at the browser/client side.
- The following are applications created with Sencha Touch:
  - *TravelMate:* Travel domain application
  - *Xero:* Mobile-based accounting application
  - *Tubetweet:* A chatting and tweeting application

Intel XDK

- The XDK framework from Intel comes with an end-to-end development studio for mobile application developers.

- It consists of almost everything required for hybrid development including IDE, debugger, emulators, and deployment helpers.

- Code is supported using HTML5 and JavaScript.

- Some of the applications created with Intel XDK are as follows:
  - *Home Remedies:* Helper-in-life kind of application
  - *Press and Hard:* A puzzle game application

Testing Mobile Applications

- Hybrid mobile application testers and developers may test applications for functionality, usability, and consistency.
- **Testing with Browsers** : If an application is created with HTML and JavaScript, that application can be tested for functionality on a browser.
- The Chrome, IE, and Firefox browsers provide developer tools to debug and simulate certain behaviors.
- The key challenge in this type of application testing is that the market has many devices with different screen sizes, operating systems, and so forth.
- How can you test applications for different mobile device sizes?
- A developer who is running the application can use **ww.responsinator.com**, to test the UI by using simulation.

Testing Mobile Applications

- Although working with a browser while testing has its advantages, you can't test device features such as an SD card or camera, because these features will be available only on mobile devices.

- Hybrid applications can access device features when run in a *web view;* this means that though these applications are running in a browser context, they will always be running with elevated permissions.

- So, you need to install applications on devices in order to test them.

- **Testing on Devices** : If you need to test hybrid applications on iOS, Windows, or Android devices, each platform vendor will have different rules.

- Apple and Microsoft demand that you register for an Apple or Microsoft developer account.

- Testing on multiple devices is costly.

Testing Mobile Applications

- **Testing with Packaging Frameworks** : While developing application IDEs such as Visual Studio, Intel XDK may help developers and testers to debug or test applications.

- For an AppBuilder-like framework, you may get an extension to an IDE like Visual Studio.

- These IDEs also come with emulators that help test the program on the dev machine itself.

- After the testing phase is over, deployment of the application is planned.

Deploying Applications

- Distributing an application in the Android ecosystem can be done freely and easily. If you want the app to be uploaded on Google Play, you must have a developer account.

- When it comes to Apple and Microsoft, you are required to have a developer license. This allows one developer to test a specified number of applications on a single device.

- If the application hosted is a paid one, then Apple, Microsoft, or Google takes a share in the minimum amount of money that you earn. Typically, the share ratio is 70:30.

Considering Cost

- If your client wants an application to be put on iOS, Windows Phone, and Android devices, this native app development will be expensive in terms of money and effort because of the various compliance restrictions.

- But when you decide to go hybrid, you have the following benefits:
  - Less cost
  - One code base
  - Responsive code that helps build the UI for each type of device
  - Less development time


THANKS