

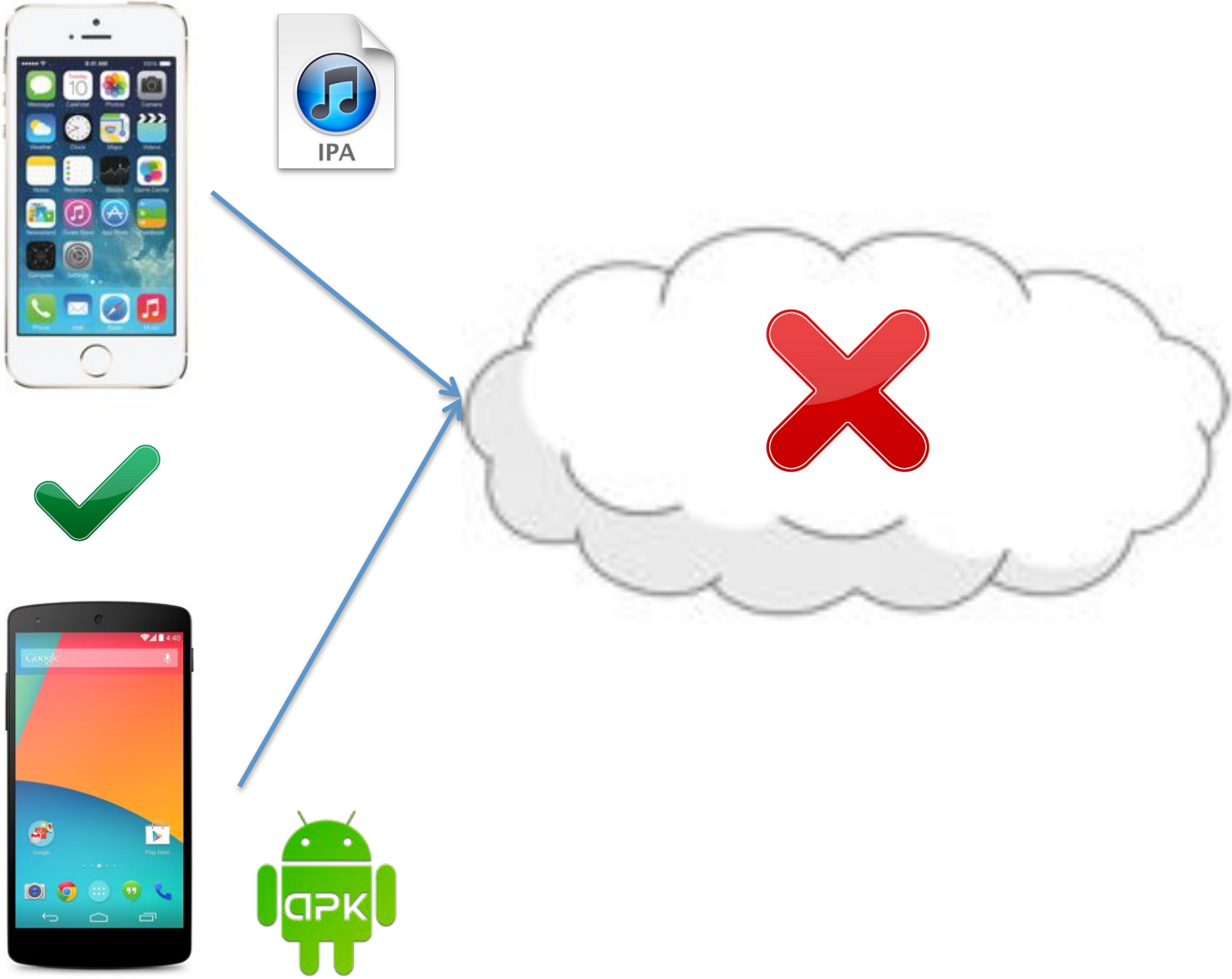
# Client-Side Mobile Architecture Choices

WebSphere User Group, Edinburgh, 30<sup>th</sup> September 2014

*Andrew Ferrier,  
Technical Lead,  
IBM Software Services for WebSphere Mobile Practice*



# How do I implement my mobile application?




# Agenda

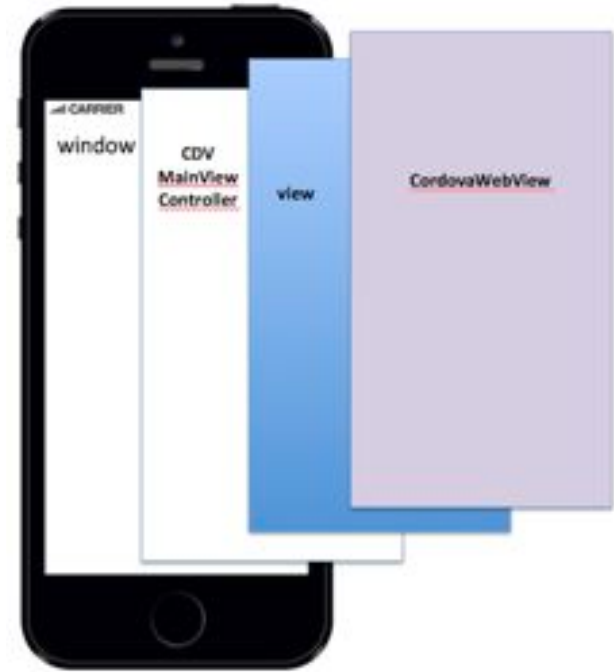
- What ways are there to write an app – and what's Cordova?
- What's IBM Worklight Foundation and how is it relevant?
- Web Technology – JavaScript Toolkits
- Native Coding – Android and iOS
- How can we mix these?
- Summary – which is best?
- How does BlueMix relate?
- How does the Apple / IBM partnership relate?

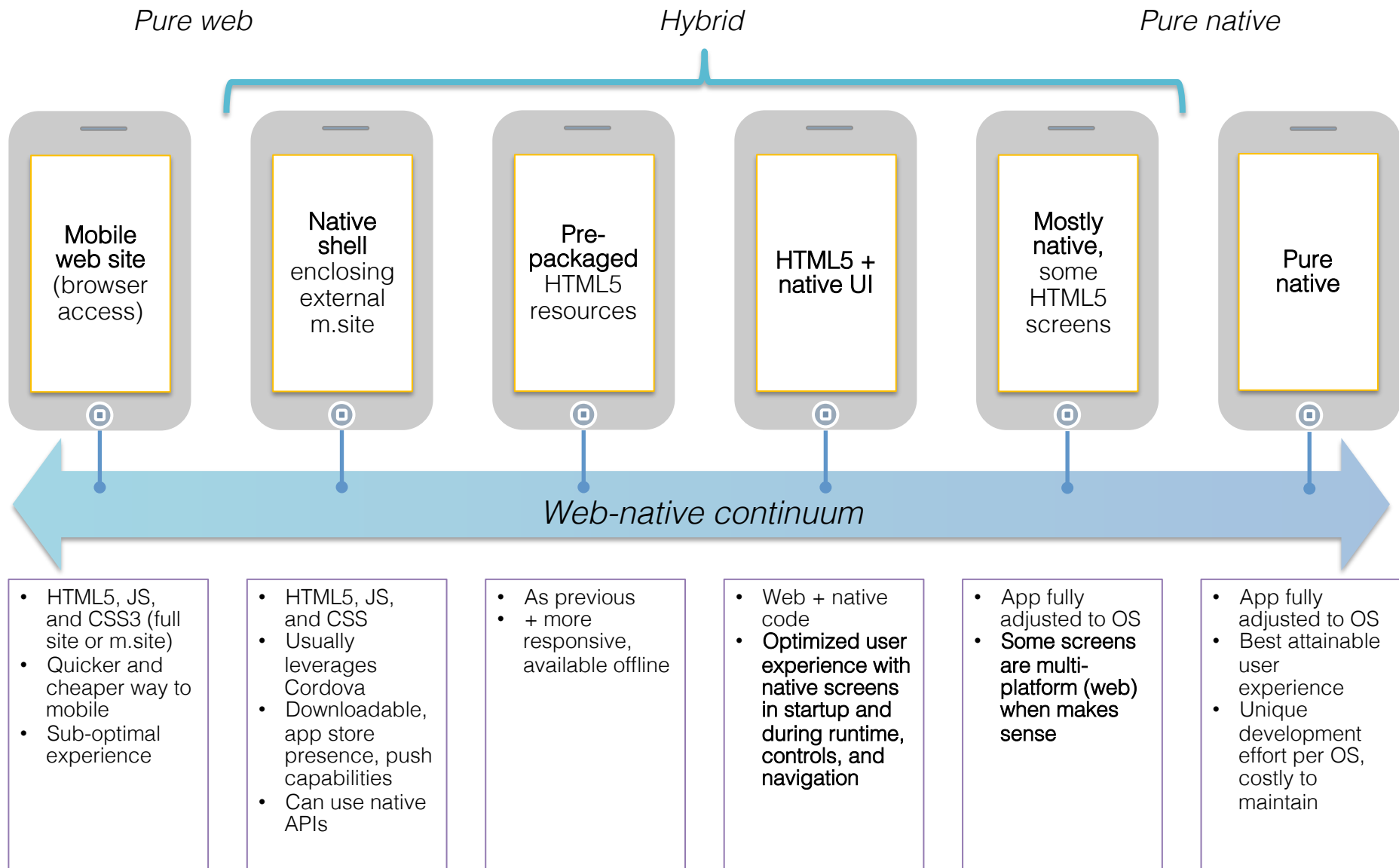
# 3 different ways to write an app...

- Web (aka Hybrid, Cordova, etc.)
  - HTML + CSS + JS
  - Need to choose JS Toolkit – typical options include AngularJS, Dojo, and jQuery.
- Android Native SDK
  - Java
- iOS Native APIs - UIKit
  - Objective-C, Swift
- (we're not considering Windows Mobile, Blackberry)

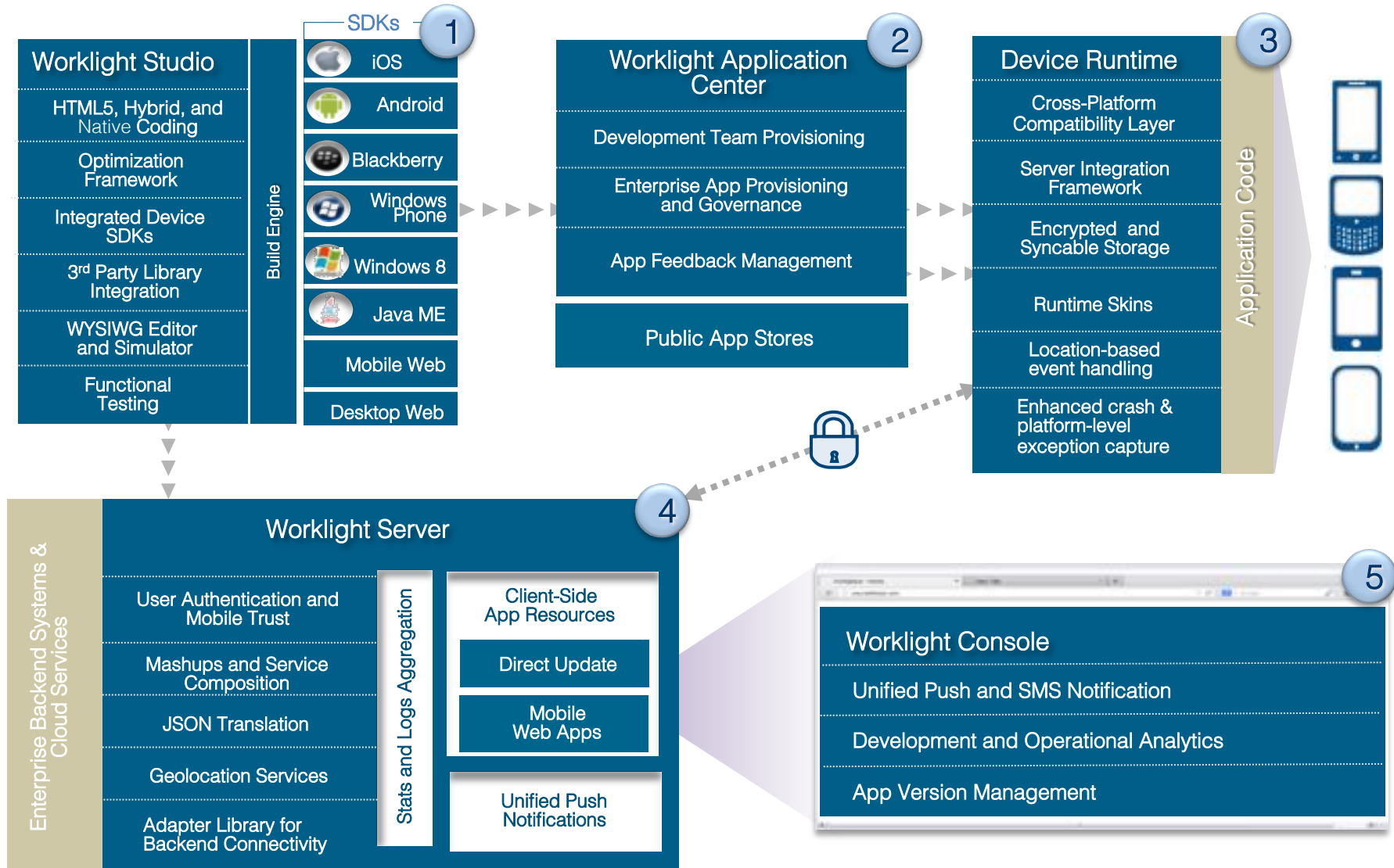
# Cordova exploits Web...

- Apache Cordova -   
<http://cordova.apache.org/>
- Previously known as PhoneGap
- Provides a wrapper around web content to make it appear as a native app
- Also provides access to device APIs

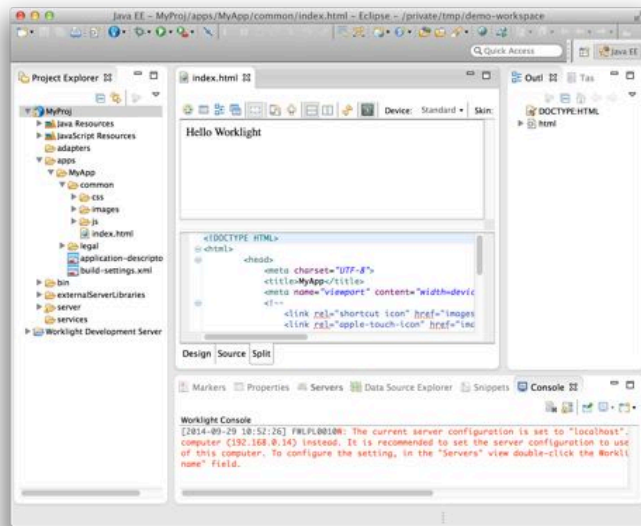




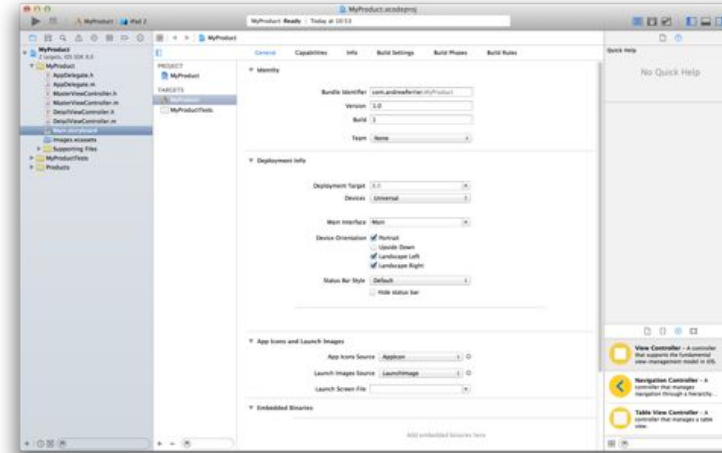
# Worklight Foundation Components



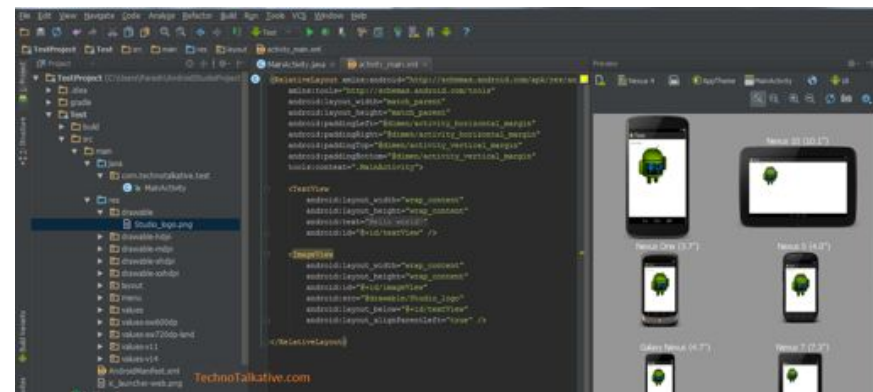
# Development Tools



Worklight IDE



Xcode

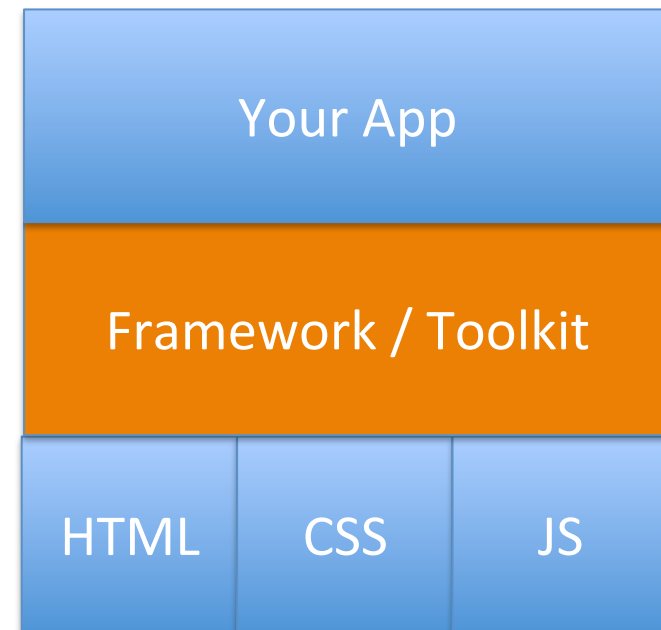


Android Studio



# Web Technologies – the Programming Model

- Coding without a JS **toolkit** or **framework** in 2014 is like entering the program in binary




# Web Technologies

- Much of what we say here might also apply to 'desktop' web applications
- The market for JavaScript toolkit choice is broadening - there are 1000s of choices, and they are changing daily.
- Split into:
  - Low-level: DOM and page architecture (aka *toolkits*)
  - High-level: Widgets (aka *frameworks*)

# Mobile Web Technologies

- AngularJS
  - + Ionic
- Dojo Mobile
- jQuery
  - + jQuery Mobile
- Sencha Touch
- Many many others...



- ‘What all the cool kids are using’
- Maintained by 
- Focus Areas:
  - One-page applications
  - MVC – strong data binding capabilities
- Not specifically focused on mobile applications, so...



- AngularJS doesn't have widgets
- Ionic adds good-looking Mobile Web-focused widgets to AngularJS
- Maintained by a startup called [drifty.com](http://drifty.com)
- Primary designed to be used declaratively
- One-single look and feel...



+



## Header Buttons

Edit

### List Icons

- Check mail
- Call Ma
- Record album Grammy
- Friends 0
- Activities**
- Breaking Bad Blue, yellow, pink
- Music JT
- Games Super Mario
- Shopping →

### Inline Labels

Username

Password

[Sign In](#)

### Toggle

- HTML5
- CSS3
- Flashplayer
- Java Applets
- JavaScript
- Silverlight
- Web Components



Crunch

Hourly	\$15.00
Daily	\$120.00
Weekly	\$600.00
Bi-Weekly	\$1,200.00
Semi-Monthly	\$1,300.00
Monthly	\$2,600.00
Annually	\$31,200.00

**Weekly Schedule**

Hours/Day 8

Days/Week 5

MALLZEE

NEW IN >

NEW IN STYLES

Realiser

Brands Settings



Rue Crémieux  
75012 Paris

Station Quai de la Rapée



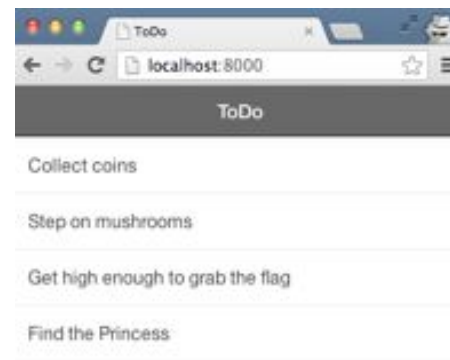
+



```
<!-- Center content -->
<ion-side-menu-content>
  <ion-header-bar class="bar-dark">
    <h1 class="title">Todo</h1>
  </ion-header-bar>
  <ion-content>
    <!-- our list and list items -->
    <ion-list>
      <ion-item ng-repeat="task in tasks">
        {{task.title}}
      </ion-item>
    </ion-list>
  </ion-content>
</ion-side-menu-content>
```

```
angular.module('todo', ['ionic'])

.controller('TodoCtrl', function($scope) {
  $scope.tasks = [
    { title: 'Collect coins' },
    { title: 'Eat mushrooms' },
    { title: 'Get high enough to grab the flag' },
    { title: 'Find the Princess' }
  ];
});
```





- AngularJS + Ionic is a good 'default' choice.
- There are established patterns and examples for using it with Worklight:
  - [http://www.youtube.com/watch?v=a89W\\_atlhjg](http://www.youtube.com/watch?v=a89W_atlhjg)
  - <https://github.com/g00glen00b/ibm-worklight-angular>
- When you want to stray from MVC, you may find it doesn't offer as much





- Open-source toolkit, maintained by Dojo Foundation, primary sponsor is SitePen
- IBM has some involvement in development – supported through the IBM Worklight product in
- Latest version (1.10.1) brings iOS 8 support

# Why **dōjō** toolkit ?

- Enterprise-grade toolkit and feature set
- Strong support for structuring large applications
  - e.g. AMD, Class system (`dojo/declare`), `dojox/app`
- Better focus on internationalization, accessibility, etc.
- Strong theming ability for mobile widgets
- Good option for multi-channel or desktop applications.

# dojo toolkit

Showcase	Form
<b>Controls</b>	<b>Personal Data</b>
Buttons <sup>1.6</sup>	Full name: <input type="text" value="Luko"/>
Forms <sup>1.6</sup>	Hide birthdate: <input checked="" type="checkbox"/>
Switches <sup>1.7</sup>	Goals: <input type="text"/>
Flippable <sup>1.6</sup>	<b>Login</b>
Icons <sup>1.6</sup>	User name*: <input type="text"/>
Tab Bar <sup>1.6</sup>	Password*: <input type="text"/>
Headings <sup>1.6</sup>	<b>Alerts</b>
Map (Google) <sup>1.6</sup>	Issue alerts for:
Lists <sup>1.6</sup>	All messages: <input type="checkbox"/>
List Data <sup>1.7</sup>	Urgent messages only: <input checked="" type="checkbox"/>
Gauge <sup>1.7</sup>	Audible alerts: <input type="checkbox"/> OFF
Chart <sup>1.7</sup>	Alert volume: <input type="range"/>
Geo Chart <sup>1.7</sup>	<input type="button" value="Reset Form"/>
OpenLayers Map <sup>1.7</sup>	
Touch <sup>1.7</sup>	
<b>Effects</b>	
Transitions <sup>1.7</sup>	

The screenshot shows the Dojo Toolkit documentation interface. On the left, a 'Calendar' widget displays the month of September. On the right, a 'Form' widget is shown with various input fields and controls. The 'Form' widget includes sections for 'Buttons', 'CheckBox', and 'Radio Buttons'. The 'Buttons' section shows examples of enabled and disabled buttons with different styles (Simple, Drop Down, Combo, Toggle). The 'CheckBox' section shows examples of unchecked, checked, disabled, and disabled and checked states. The 'Radio Buttons' section shows examples of news, talk, and weather (disabled) options. The 'Form' widget also includes a 'Reset Form' button at the bottom.

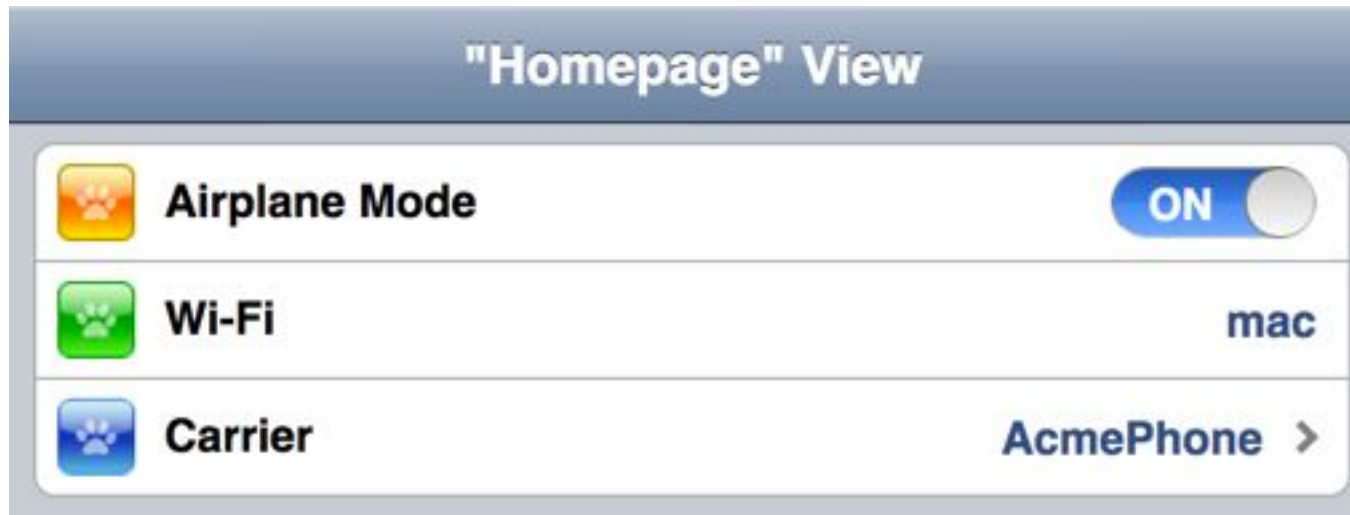
# dojox/mobile

- Part of the Dojo toolkit focused on mobile applications
- Core concept is dojox/mobile/Views
  - Dynamic loading of Content
- Mobile widgets in dojox/mobile/\*
- But OOB LnF somewhat dated (iOS 6 era)




# dojox/mobile

```
<!-- the view or "page"; select it as the "home" screen -->
<div id="settings" data-dojox-type="dojox/mobile/View" data-dojox-props="selected:true">
  <!-- a sample heading -->
  <h1 data-dojox-type="dojox/mobile/Heading">"Homepage" View</h1>
  <!-- a rounded rectangle list container -->
  <ul data-dojox-type="dojox/mobile/RoundRectList">
    <!-- list item with an icon containing a switch -->
    <li data-dojox-type="dojox/mobile/ListItem" data-dojox-props="icon:'images/icon-1.png'">Airplane Mode
      <!-- the switch -->
      <div data-dojox-type="dojox/mobile/Switch"></div>
    </li>
    <!-- list item with an icon that slides this view away and then loads another page -->
    <li data-dojox-type="dojox/mobile/ListItem" data-dojox-props="icon:'images/icon-2.png', rightText:'mac'">Wi-Fi</li>
    <!-- list item with an icon that slides to a view called "general" -->
    <li data-dojox-type="dojox/mobile/ListItem" data-dojox-props="icon:'images/icon-3.png', rightText:'AcmePhone', moveTo:'general'">Carrier</li>
  </ul>
</div>
```





- The core of jQuery is lightweight and simple, and is the most popular framework.
- Owned and run by jQuery Foundation, IBM a Founding Member
- It is **NOT**   
JavaScript
- It is used by AngularJS (either in full or stripped-down form)
- Similar to AngularJS in that it doesn't have its own widget library

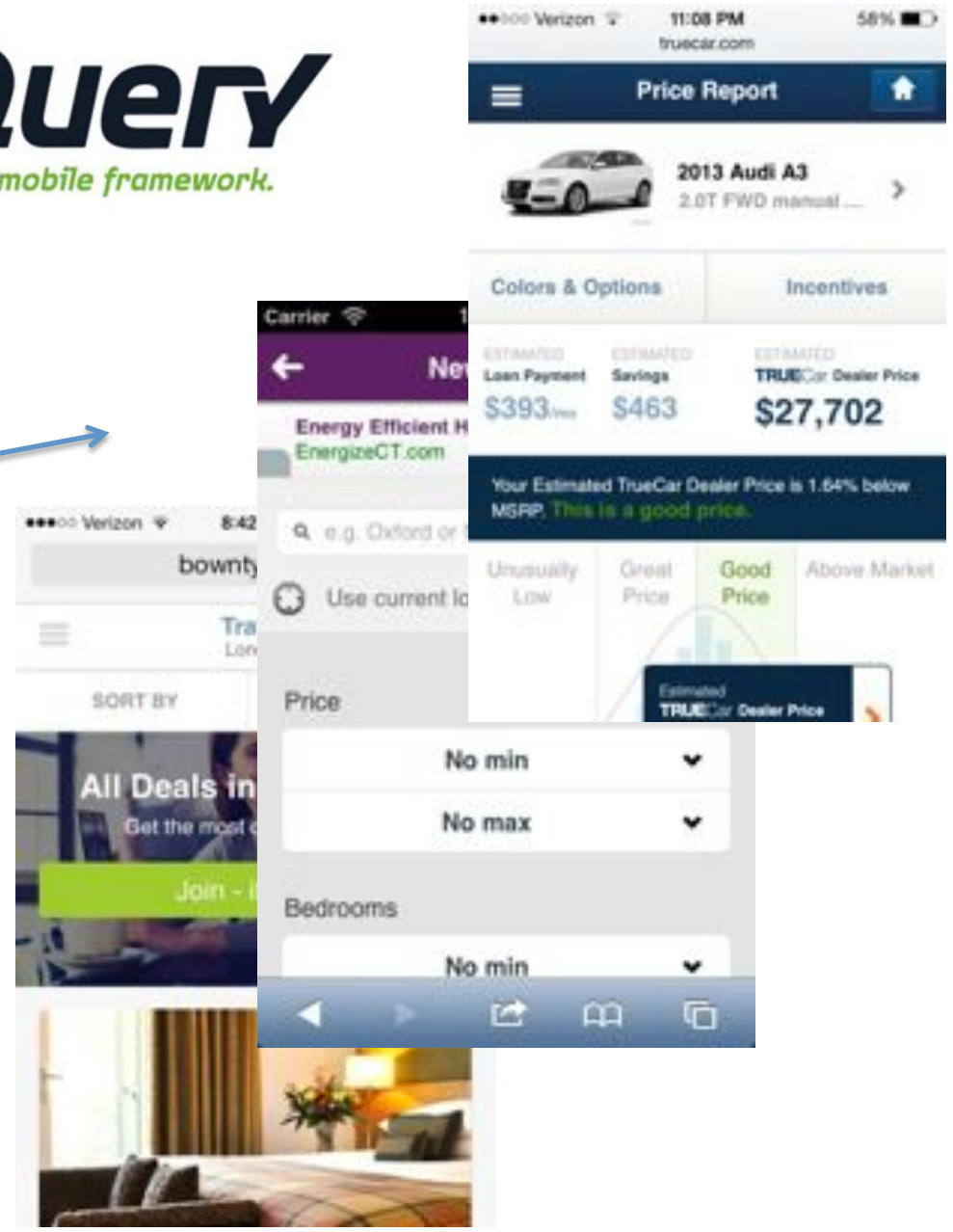
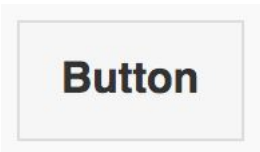


- Focuses on 'core' functionality:
  - DOM traversal and manipulation
  - Event Handling
  - AJAX / XHR
- UI Widgets, Unit Testing, etc. all separate projects:





Mobile Framework also separate:







- No real MVC support
  - Although addons, e.g. JMVC  
<http://www.javascriptmvc.com/>)
- Simple to get started
- Not themed towards any particular mobile OS
- Because jQuery is a very lightweight framework, doesn't box you in
  - Advantages and disadvantages



# Sencha Touch / ExtJS

- Sencha Touch is a mobile widget library
- Built off ExtJS, a more generic JS library
- Situation akin to AngularJS + Ionic
- Owned by Sencha company, more product-oriented
- MVC-oriented
- Harder to combine with other toolkits
- Sencha Touch is monolithic, library is large
- No declarative HTML...



# Sencha Touch / ExtJS

```
Ext.define('Twitter.view.SearchBar', {
    extend: 'Ext.Toolbar',
    xtype: 'searchbar',
    requires: ['Ext.field.Search'],

    config: {
        ui: 'searchbar',
        layout: 'vbox',
        cls: 'big',

        items: [
            {
                xtype: 'title',
                title: 'Twitter Search'
            },
            {
                xtype: 'searchfield',
                placeholder: 'Search...'
            }
        ]
    }
});
```



# Sencha Touch / ExtJS

**TouchStyle**

*touch the latest styles*

<p><b>Clothing</b> Denim, Dresses, Intimates, Jackets, Outerwear, Pants &amp; Shorts, Skirts,</p>	<p><b>Shoes</b> Athletic, Boots, Evening, Flats, Mules &amp; Clogs, Platforms, Pumps, Sandals,</p>	<p><b>Handbags</b> Backpacks, Clutches, Evening, Hobos, Satchels, Shoulder Bags,</p>	<p><b>Accessories</b> Belts, Hats, Gloves &amp; Wraps, Sunglasses,</p>
<p><b>Beauty</b> Bath &amp; Body, Fragrance, Hair Accessories, Haircare, Makeup, Skin</p>	<p><b>Men</b> Accessories, Jeans, Outerwear, Pants, Shirts, Shoes, Shorts,</p>	<p><b>Kids &amp; Baby</b> Newborn (0-12m), Infant Boys (0-24m), Infant Girls (0-24m), Toddler Boys (2T-3T)</p>	<p><b>Home &amp; Living</b> Bed &amp; Bath, Decor, Furniture, Kitchens, Lighting, Rugs, Tabletop</p>

Web 2.0 Summit 2010		
11/15	11/16	11/17
12:00pm		
Innovation: New Ideas from Promising Start-ups		<a href="#">See Cliff</a>
Is the PC Dead? Re-imagining Cloud Computing for the Next Six Billion		<a href="#">Marina</a>
Q&A: The New Search Insurgent		<a href="#">Pacific Heights</a>
2:00pm		
A Conversation with Eric Schmidt, CEO, Google		<a href="#">Grand Ballroom</a>
Opening Welcome		<a href="#">Grand Ballroom</a>
3:00pm		
Playing for High Stakes in the New App Economy		<a href="#">Grand Ballroom</a>
Point of Control: Commerce		<a href="#">Grand Ballroom</a>
Where Good Ideas Come From		<a href="#">Grand Ballroom</a>
4:00pm		
Ignite: Internet Culture: What Does It Mean?!?!		<a href="#">Grand Ballroom</a>
Point of Control: Consumer Platforms		<a href="#">Grand Ballroom</a>
The Network Is The Platform For Transforming The World		<a href="#">Grand Ballroom</a>
5:00pm		
<a href="#">Sessions</a> <a href="#">Speakers</a> <a href="#">Tweets</a> <a href="#">Location</a> <a href="#">About</a>		

# For more comparisons...



The image shows a screenshot of the TodoMVC website. At the top left is the 'TodoMVC' logo, which includes a red checkmark inside a dashed square, followed by the text 'TodoMVC' in a bold, sans-serif font. Below the logo is the tagline 'Helping you select an MV\* framework'. To the right of the tagline are three buttons: 'Download', 'View on GitHub', and 'Blog'. On the top right of the page is an illustration of a clipboard with a red checkmark and a red pen.

The main content area is divided into several sections:

- Introduction:** A paragraph explaining the purpose of the site, which is to help developers choose between different MV\* frameworks. It mentions that the site is a collection of examples and is not a framework itself.
- JavaScript Apps:** A grid of links to various JavaScript applications that use different frameworks. The links are arranged in four columns and several rows. Some links are highlighted in red.
- Compile To JavaScript:** A section listing frameworks that can be compiled to JavaScript, such as CoffeeScript and TypeScript.

# Consider your tools...



- Bower - Package Management – [bower.io](http://bower.io)



- Grunt – ‘Task Runner’ – [gruntjs.com](http://gruntjs.com)






- KARMA - ‘Test Runner’ – [karma-runner.github.io](https://github.com/karma-runner/karma)

# Native Technologies

- iOS – for iPhone/iPod, iPad, Watch
  - Objective-C
  - Swift
- Android – for Phone, Tablet, Google Glass, Watch
- Windows Phone
- Blackberry

# iOS Native

- On iOS, you broadly have two technology choices:
  - Objective-C – Older, harder to learn  
  - Swift – Only introduced this year. 
- The two can be combined.

Objective-C:

```
Person *matt = [[Person alloc] initWithName:@"Matt Galloway"];  
[matt sayHello];
```

Swift:

```
var matt = Person(name:"Matt Galloway")  
matt.sayHello()
```

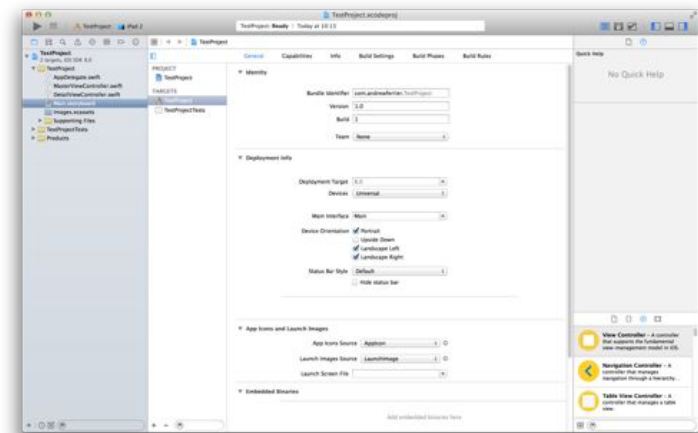


# iOS Native

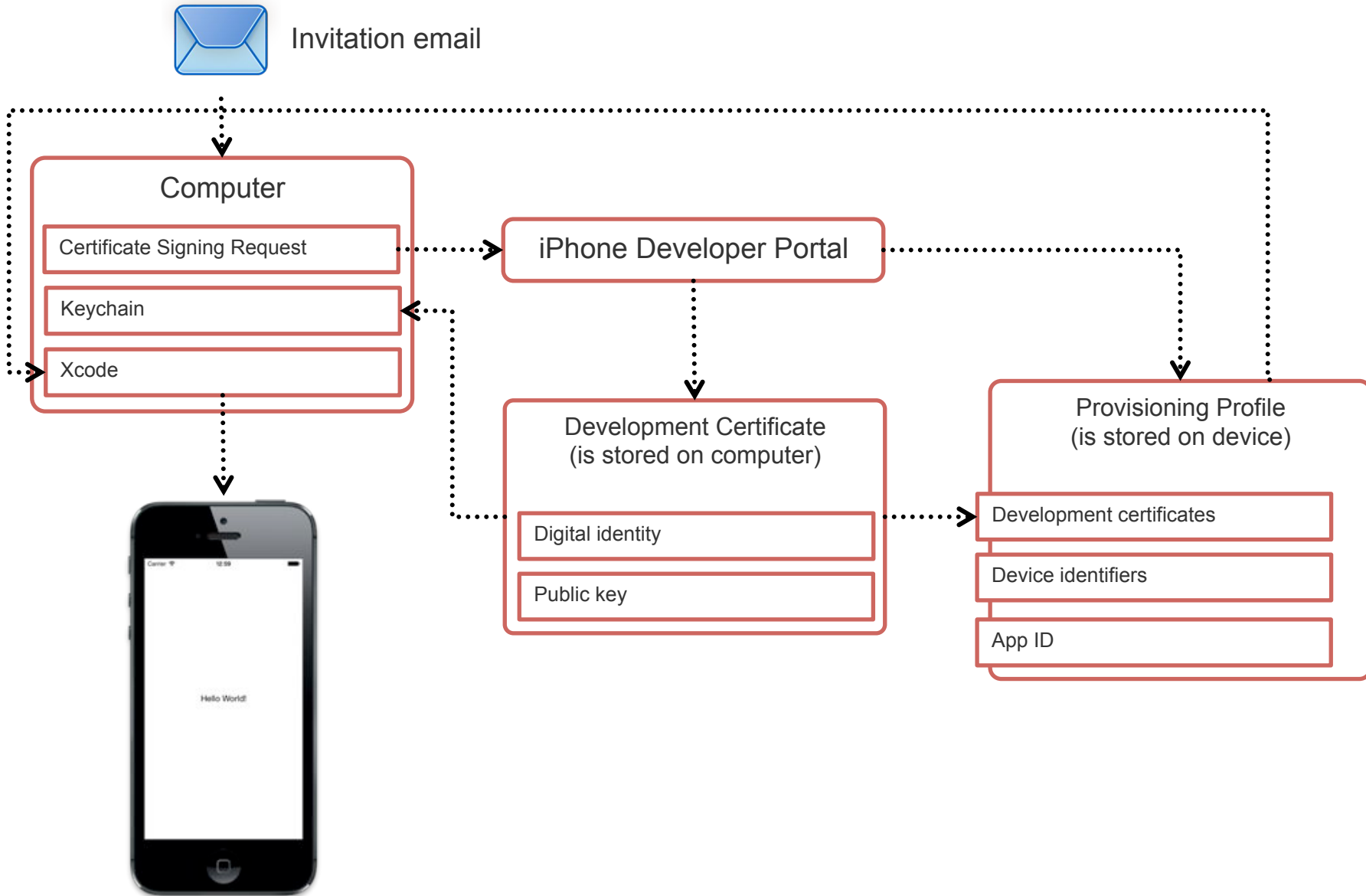
- For all apps (not just native), you need to **register** to deploy to “real” devices



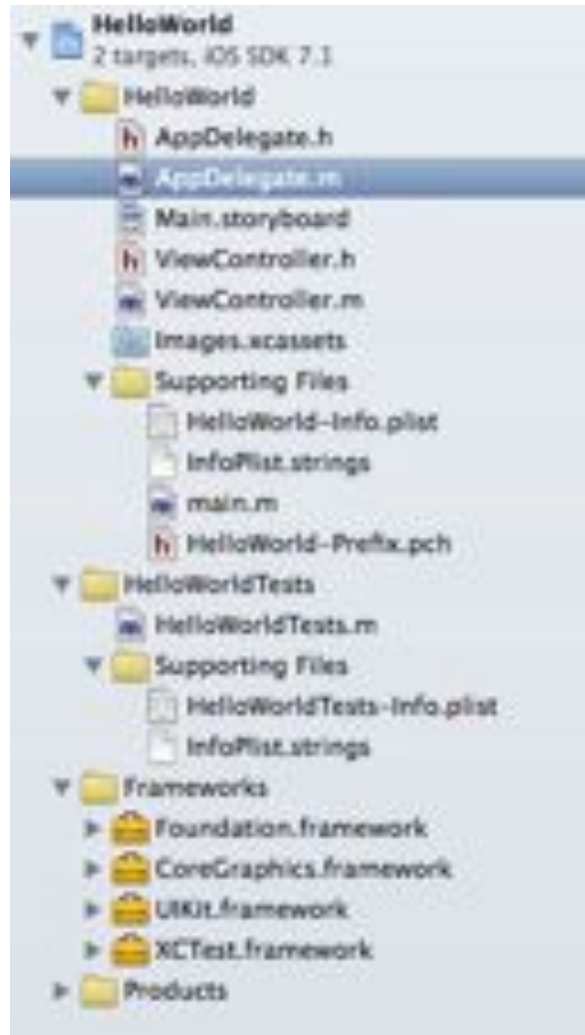
- Also need Xcode development environment (***only*** supported on Mac)



# Steps for the developer

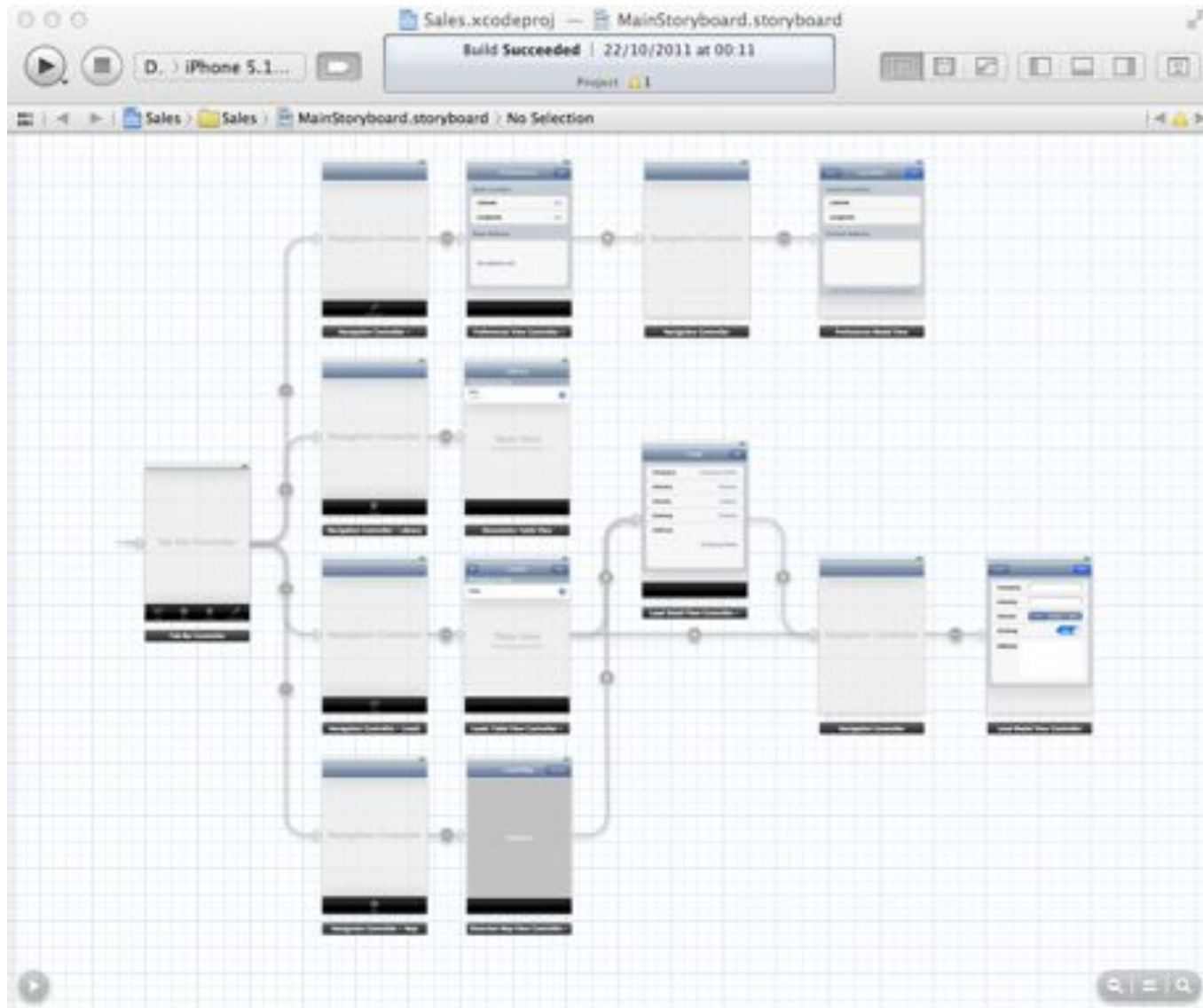


# Xcode project structure



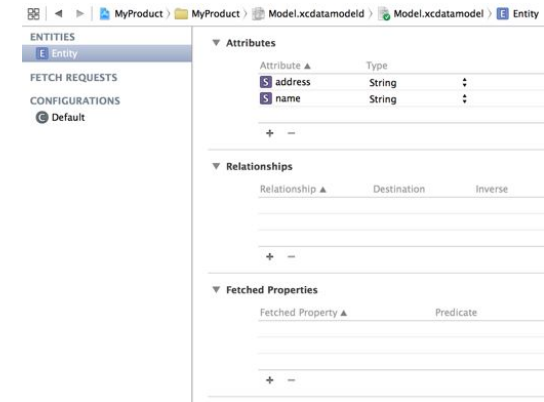
- **Project file “HelloWorld”**
  - Comparable to a manifest file
  - General information about the app
  - Capabilities selection
  - Build configuration
- **\*.h Header files** – public interface of a class
- **\*.m Implementation files** incl. private interface
- **AppDelegate**
  - Handles lifecycle event of the app
  - Is a global implementation file
  - Can be called from all implementation files
- **\*.storyboard** – Definition of the user interface using Interface Builder
- **\*.xib** – Definition of a single view using Interface Builder
- **Images.xcassets** – collection of image resources for different resolutions
- **Supporting files** – misc. files, e.g. localized resources
- **HelloWorldTests** – definition of test cases
- **Frameworks** – included platform libraries
- **Products** – actual distribution files

# Storyboarding



# Provides MVC...

- **View** - Components created using Interface Builder (Storyboards / XIBs)
- **Model** - Objective-C classes or Core Data



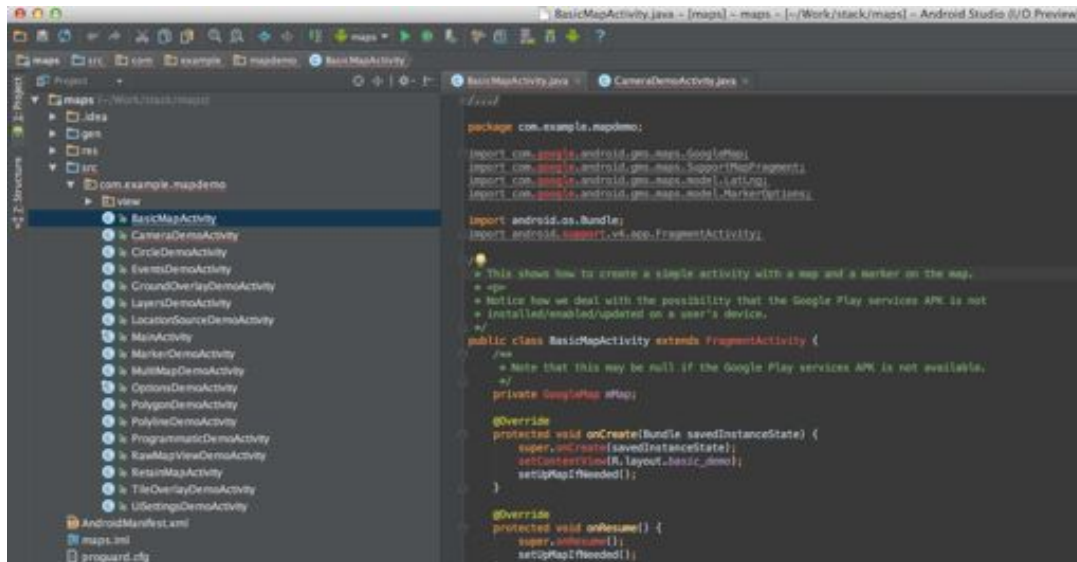
- **Controller** – Typically, these subclass existing generic controller classes from the UIKit framework such as UIViewController

# Objective-C vs. Swift

- Both can use the same APIs
- Can be mixed inside an application
- Swift has many advantages – easier, safer, better language features
- But for now, you still need to know Objective-C:
  - Much of the community still talks in Objective-C
  - Most frameworks (including WL's) are written in Objective-C, so debugging?
- There is also a de-facto package manager in the form of CocoaPods (<http://cocoapods.org/>)

# Android Native

- Install Android Studio (<http://developer.android.com/sdk/installing/studio.html>) - based on IntelliJ
- Eclipse plugins will be phased out



```
BasicMapActivity.java - [maps] - maps - [~/Work/stack/maps] - Android Studio (I/O Preview)
maps | src | com | example | mapdemo | BasicMapActivity
Project
  maps
  |__ src
  |   |__ com
  |       |__ example
  |           |__ mapdemo
  |               |__ view
  |                   |__ BasicMapActivity
  |                   |__ CameraDemoActivity
  |                   |__ CircleDemoActivity
  |                   |__ EventsDemoActivity
  |                   |__ GroundOverlayDemoActivity
  |                   |__ LayersDemoActivity
  |                   |__ LocationSourceDemoActivity
  |                   |__ MainActivity
  |                   |__ MarkerDemoActivity
  |                   |__ MultiMapDemoActivity
  |                   |__ OptionsDemoActivity
  |                   |__ PolygonDemoActivity
  |                   |__ PolylineDemoActivity
  |                   |__ ProgrammaticDemoActivity
  |                   |__ RawMapViewDemoActivity
  |                   |__ RetainMapActivity
  |                   |__ TileOverlayDemoActivity
  |                   |__ UGSettingsDemoActivity
  |__ AndroidManifest.xml
  |__ maps.iml
  |__ proguard.cfg
BasicMapActivity.java
CameraDemoActivity.java
//...
package com.example.mapdemo;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import android.os.Bundle;
import android.support.v4.app.FragmentActivity;

//...
// This shows how to create a single activity with a map and a marker on the map.
// -*-
// Notice how we deal with the possibility that the Google Play services APK is not
// installed/enabled/updated on a user's device.
//...
public class BasicMapActivity extends FragmentActivity {
    //...
    // Note that this may be null if the Google Play services APK is not available.
    //...
    private GoogleMap mMap;

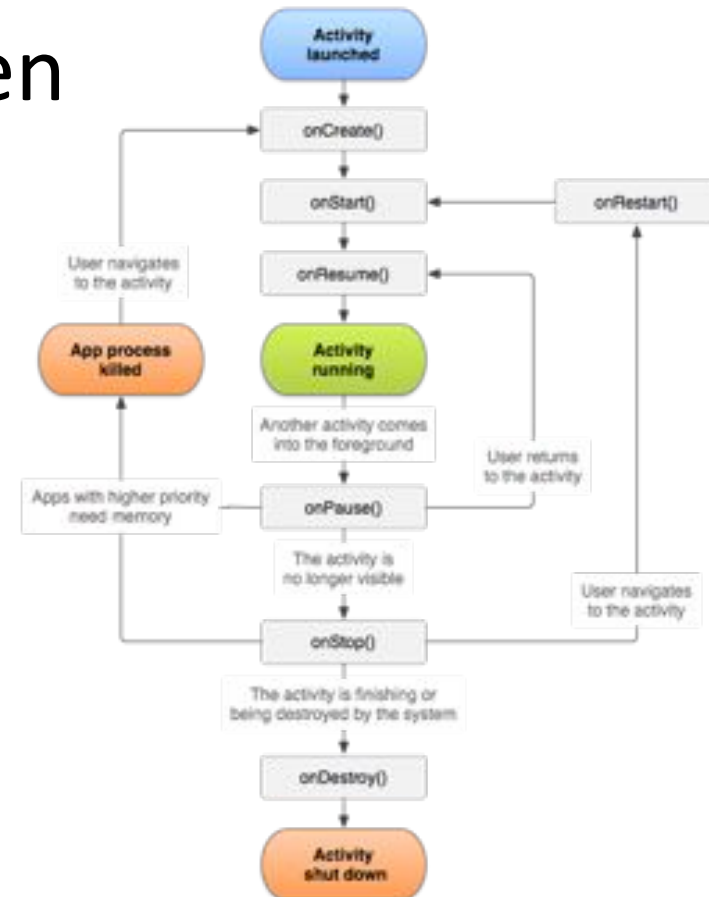
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.basic_demo);
        setUpMapIfNeeded();
    }

    @Override
    protected void onResume() {
        super.onResume();
        setUpMapIfNeeded();
    }
}
```



# Activities

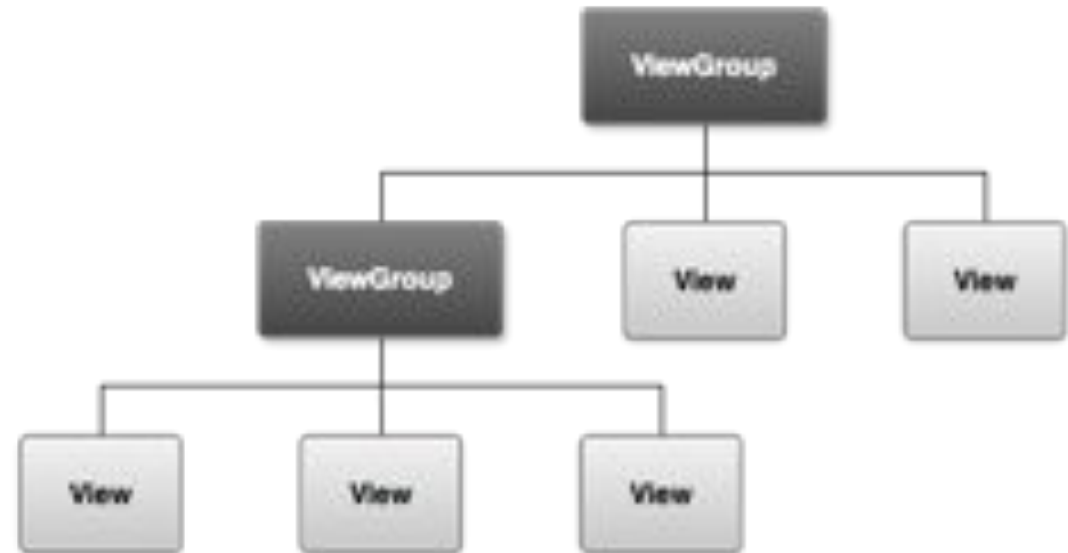
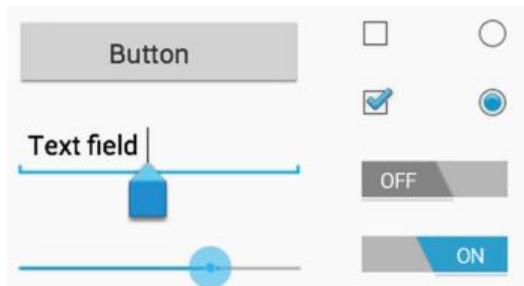
- Typically a Single screen
- Stack of Activities
- Can be persisted





# ViewGroups and Views

- ViewGroup  $\sim$  Container
- View  $\sim$  Widget (aka Input Controls):



# Programmatic or Declarative

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_text"  
    ... />
```

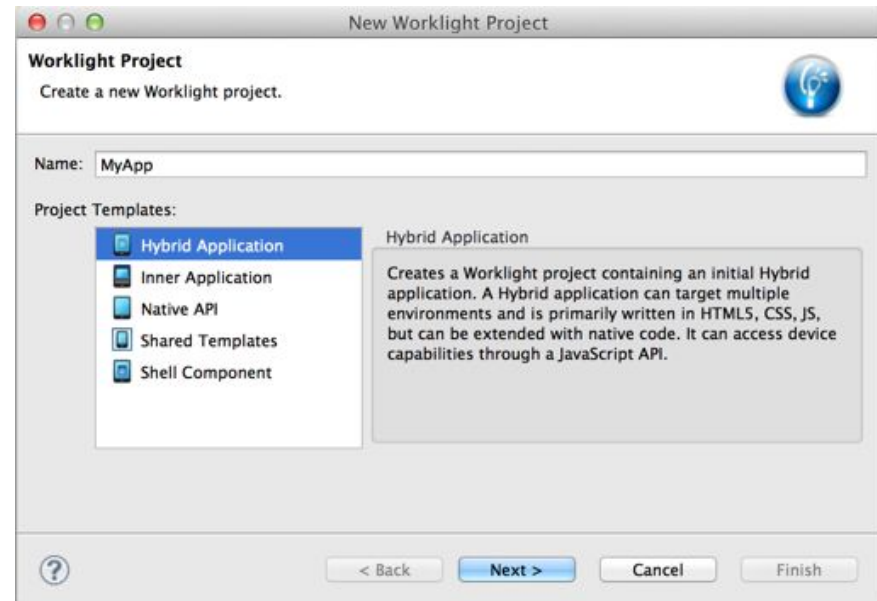


Alarm

```
LinearLayout myLayout = findViewById(R.id.main);  
  
Button myButton = new Button(this);  
myButton.setLayoutParams(new LinearLayout.LayoutParams(  
    LinearLayout.LayoutParams.FILL_PARENT,  
    LinearLayout.LayoutParams.FILL_PARENT));  
  
myLayout.addView(myButton);
```

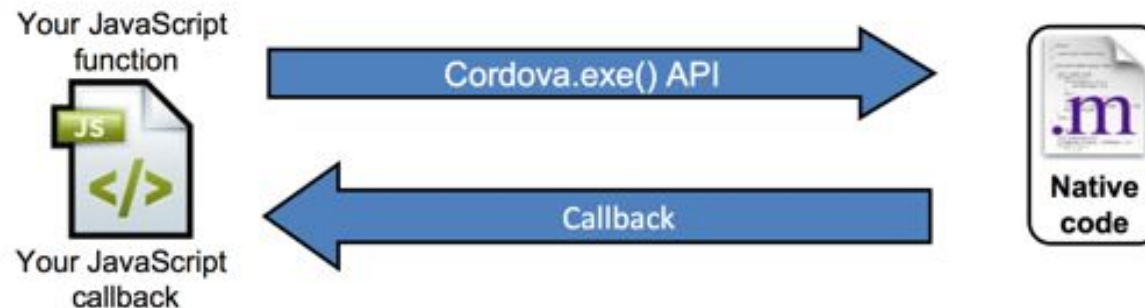
# How can we mix all of these?

- Two basic styles:
  - **Hybrid** web container is master - in Worklight, this is a **Hybrid** application (i.e. Cordova)
  - **Native** code is master - in Worklight, this is a **Native API** application



# Adding native functionality to **hybrid** apps

- To create and use an iOS Cordova plug-in:
  - Declare the plug-in in the config.xml file.
  - Use cordova.exec() API in the JavaScript code.
  - Create the plug-in class that will run natively in iOS.
- The plug-in performs the required action, and calls a JavaScript callback method that is specified during the cordova.exec() invocation.



# Adding native pages to **hybrid** apps

- Use the `WL.NativePage.show()` API to start a native page:

```
function openNativePage(){  
    var params = {nameParam : $('#nameInput').val()};  
    WL.NativePage.show('HelloNative', backFromNativePage, params);  
}  
  
function backFromNativePage(data){  
    alert("Received phone number is: " + data.phoneNumber);  
}
```

The name of a  
native iOS  
UIViewController  
instance to start

- (In Android, the native page is an activity.)

# Using Worklight APIs in **native** projects

- You can directly invoke Worklight's API in native iOS and Android apps. Steps:
  - Create a Worklight Native API project in Worklight Studio
  - Add pregenerated libraries & config files to your Xcode/Android project
  - Interact with the Worklight Client Singleton Object
- You can also embed your own Cordova WebViews (<http://tinyurl.com/k7bxc4x>)

# Sending actions and data between native and web

- A unified API is provided to ease mixing of JS and Native code

- `sendActionToJS (action, data)`

- `sendActionToNative (action, data)`

- `WLActionReceiver.onActionReceived (action, data)`

- `addActionReceiver (myReceiver)`

- `removeActionReceiver (myReceiver)`

```
function logoutButtonClicked(event){  
    WL.App.sendActionToNative("LogoutButtonClicked");  
}
```

```
-(IBAction)accountsButtonClicked:(id)sender{  
    NSDictionary *data = [NSDictionary dictionaryWithObject:@"MyAccounts" forKey:@"buttonId"];  
    [[WL sharedInstance] sendActionToJS:@"sideMenuButtonClicked" withData:data];  
}
```

# Summary – so which is best?

- AngularJS + Ionic is a good default choice for those with web skills
  - Maximises cross-platform compatibility
  - Disadvantages: tuning, Android variants
- iOS Native and Android Native are best for those looking for the most sophisticated cutting-edge UX
  - Typically requires more in-depth, specialised skill.









# Afterthought - How does this all relate to Bluemix?

- It doesn't, directly, but...
- Bluemix is a PaaS offering providing back-end services in the cloud to support mobile apps



# Afterthought - How does this all relate to Bluemix?

- There are APIs to communicate with Bluemix in iOS Native, Android Native, and JS Code

 <b>Get SDK zip</b> Get SDK with Gradle <b>Further reading:</b> Building Android apps Android SDK Documentation	 <b>Get SDK zip</b> Get SDK with CocoaPods <b>Further reading:</b> Building iOS apps iOS SDK Documentation	 <b>Get SDK zip</b> Get SDK with Bower <b>Further reading:</b> Building web apps JavaScript SDK Documentation
 <b>Get SDK zip</b> Get SDK with NPM <b>Further reading:</b> Building Node.js Node.js SDK Documentation	 <b>Get SDK zip</b> Get Worklight SDK plug-ins <b>Further reading:</b> Building hybrid apps Hybrid SDK Documentation	 <b>Get SDK zip</b> Get Cordova SDK plug-ins <b>Further reading:</b> Building hybrid apps Hybrid SDK Documentation

# Afterthought - How does this all relate to Apple-IBM?

- It doesn't – not directly.
- Everything we mentioned today is applicable today.
- But...

# Thanks!

# Questions?

[ibmmobiletipsntricks.com](http://ibmmobiletipsntricks.com)