Introduction to
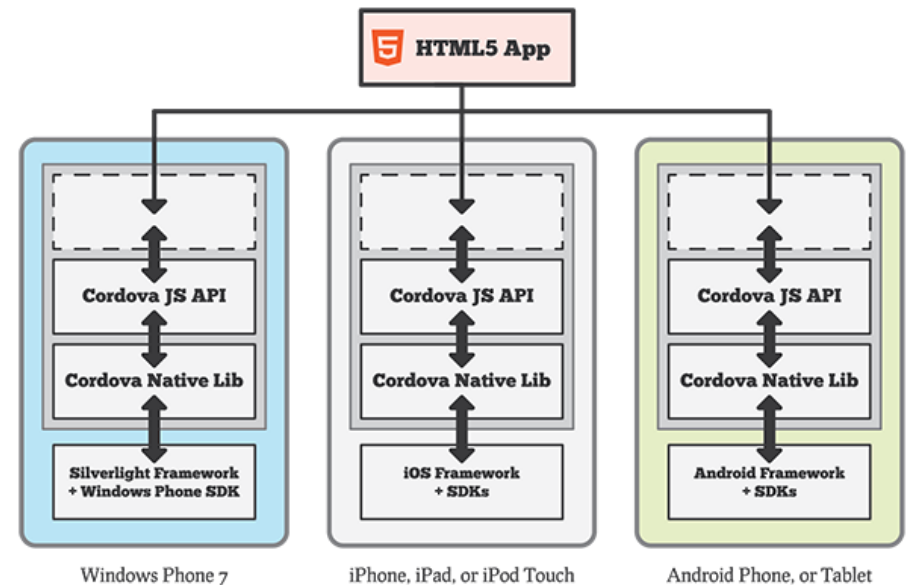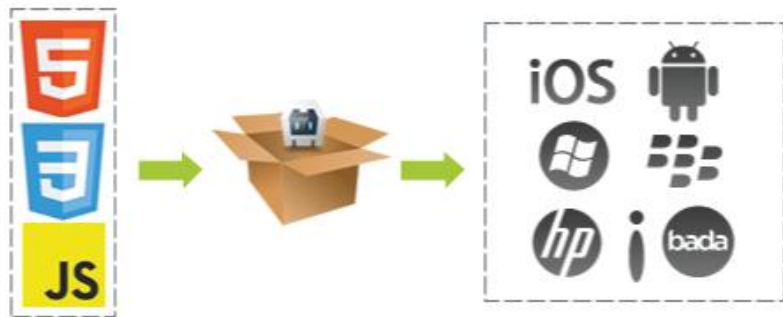

APACHE
CORDOVA™

Apache Cordova

* **Apache Cordova** is a platform for building natively installed mobile applications using **HTML, CSS** and **JavaScript**

History

* Apache Cordova was originally called **Phonegap** build by Nitobi

* Open-source & free software from the beginning (MIT License), Apache License now

* Nitobi then aquired by Adobe and donated the PhoneGap codebase to the Apache Software Foundation (ASF)

* PhoneGap is still a product of Adobe. It is **a distribution of Apache Cordova**. Think of Apache Cordova as the engine that powers PhoneGap.

Phone**Gap**

APACHE
CORDOVA™

Cordova Architecture
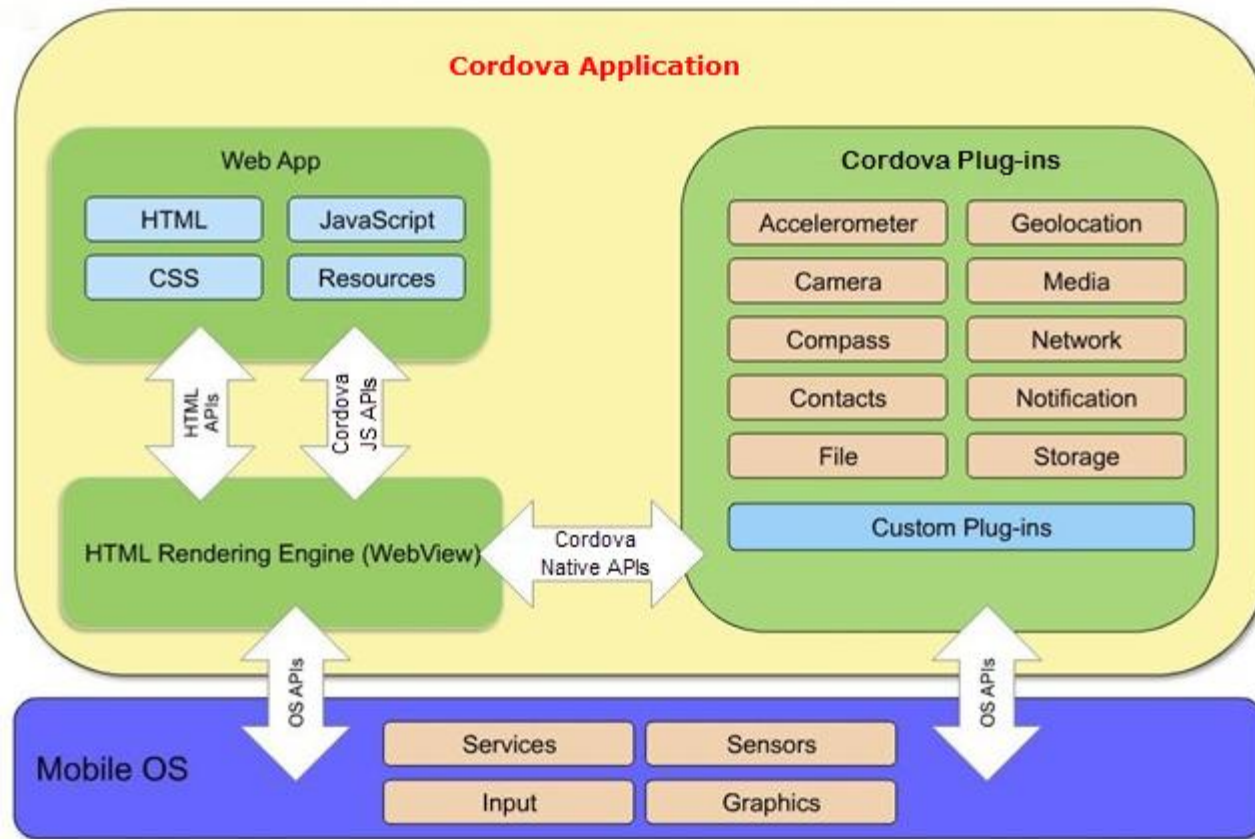
Apache Cordova Application's User Interface

* The user interface for Apache Cordova applications is created using HTML, CSS, and JavaScript.

* The UI layer is a web browser view that takes up 100% of the device width and 100% of the device height.

* The web view used by application is the same web view used by the native operating system
  * iOS: **Objective-C UIWebView class**
  * Android: **android.webkit.WebView**
  * WP7: **WebBrowser**
  * WP8: WebBrowser control (Internet Explorer 10)
  * BlackBerry: **WebWorks framework**

This is your user interface.
It is a web browser instance.
100% width & 100% height.
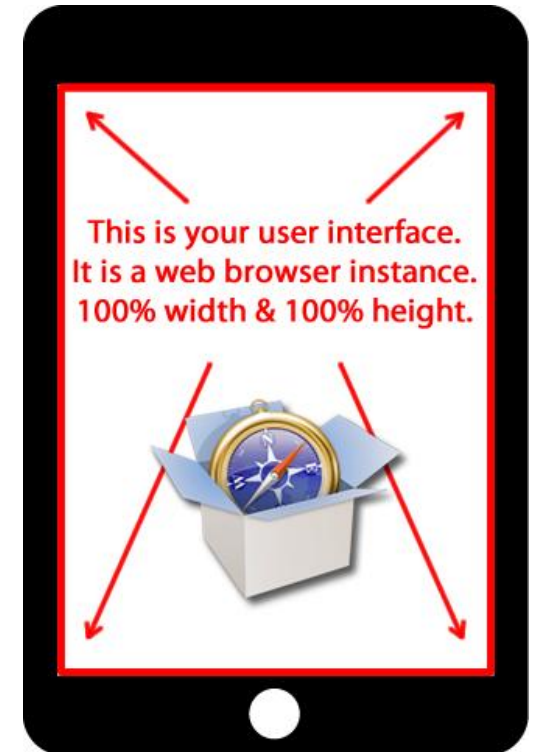
Apache Cordova API

* Provides an application programming interface (API)
  * enables you to **access native operating system functionality using JavaScript**.
  * APIs for Accelerometer, Camera, Compass, Media, FileSystem, etc
  * Extendable using native plug-in
* docs.phonegap.com

Cordova Application

Cordova JavaScript API
and
Cordova Native Library

Native API
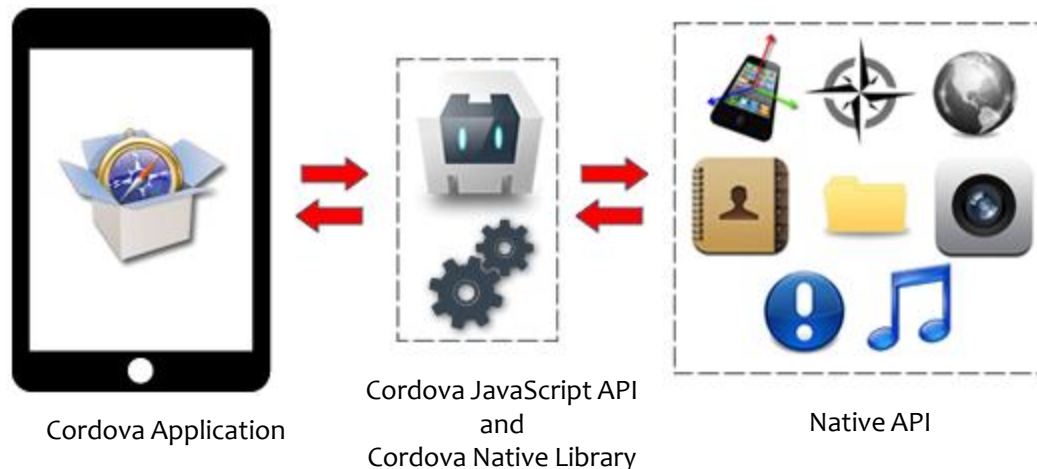
Supported Platforms

Accelerometer
Monitor the motion sensor on the device.

Camera
Take pictures with the device camera
allow the user to select images from their photo library on the device.

Capture
Capture video and still images from the camera, and audio from the microphone.

Compass
Give users of your app some direction.

Contacts
Search and Create contacts in the user's address book.

File
Low level read and write access to the file system.
Upload and download files from a web server.

GeoLocation
Make your app location aware.

Media
Play and record audio files.

Network
Monitor the device connections

Notification
Access to vibration, beep and alerts.

Storage
Persistent data store in WebStorage.

	iOS iPhone / iPhone 3G	iOS iPhone 3GS and newer	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	WebOS	WP7	Symbian	Bada
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
FILE	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
STORAGE	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗

Updated list:
http://wiki.apache.org/cordova/PlatformSupport

Development using Cordova

* ## Tools for development
  * Any HTML & JS editor
  * Platform SDK e.g. Android SDT, Android SDK, BB SDK, Xcode, Visual Studio Mobile.
  * Platform Emulator (usually provide along with SDK)
  * JS/HTML GUI Mobile framework e.g. JQuery, Sencha Touch, dojo Mobile
  * Browser e.g. Firefox with Bugzilla extension, Chrome Browser

Getting Started

Guides:

- [Getting Started with Android](#)
- [Getting Started with Blackberry](#)
- [Getting Started with iOS](#)
- [Getting Started with Symbian](#)
- [Getting Started with WebOS](#)
- [Getting Started with Windows Phone](#)
- [Getting Started with Windows 8](#)
- [Getting Started with Bada](#)
- [Getting Started with Tizen](#)
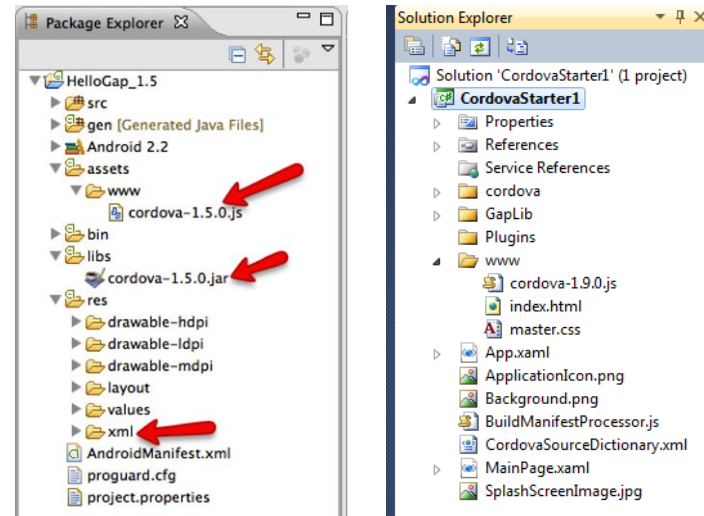
http://docs.phonegap.com/en/2.2.0/guide_getting-started_index.md.html

Use platform SDK to develop application for each target platform

				...
Xcode	Android SDK Eclipse ADT Plug-in	BB Java Eclipse Plug-in Ripple	Visual Studio, Windows Phone Dev Tools	

Code Example

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Device Properties Example</title>
    <script type="text/javascript" src="cordova-2.0.0.js"></script>
    <script type="text/javascript">
    // Wait for Cordova to load
    document.addEventListener("deviceready", onDeviceReady, false);

    // Cordova is ready
    function onDeviceReady() {
        navigator.geolocation.getCurrentPosition(onSuccess, onError);
    }

    // onSuccess Geolocation
    function onSuccess(position) {
        var element = document.getElementById('geolocation');
        element.innerHTML = 'Latitude: '          + position.coords.latitude        + '<br />' +
                            'Longitude: '         + position.coords.longitude       + '<br />' +
                            'Altitude: '          + position.coords.altitude        + '<br />' +
                            'Accuracy: '          + position.coords.accuracy        + '<br />' +
                            'Altitude Accuracy: ' + position.coords.altitudeAccuracy + '<br />' +
    }

    // onError Callback receives a PositionError object
    function onError(error) {
        alert('code: ' + error.code + '\n' + message: ' + error.message + '\n');
    }
    </script>
  </head>
  <body>
    <p id="geolocation">Finding geolocation...</p>
  </body>
</html>
```
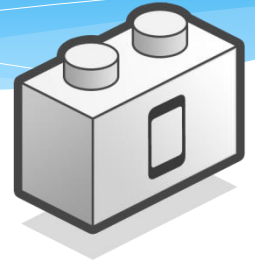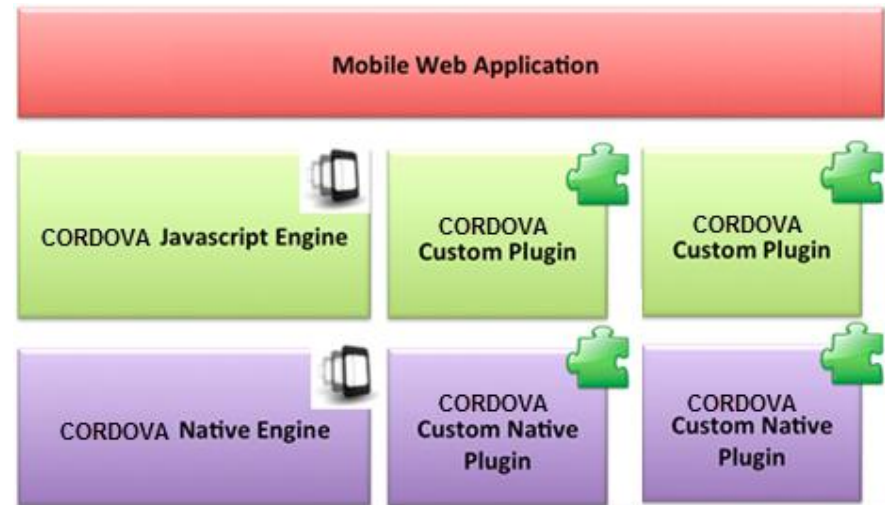
Apache Cordova Native Plug-in

What if a native feature isn't available in Core APIs?

PhoneGap is extensible with a "native plugin" model that enables you to write your own native logic to access via JavaScript.

* You develop your JavaScript class to mirror the API of the native class
* Invoke the native function using `PhoneGap.exec()`
* Plug-in class mappings:
  * **Android:** res/xml/plugins.xml
  * **iOS:** www/Cordova.plist
  * **BlackBerry:** www/plugins.xml


```
PhoneGap.exec(function(winParam){}, function(error){}, "service", "action", [params]);
```

Plugin Example (Android Native Code)

```java
package sample.cordova.plugin;

import org.apache.cordova.api.CordovaPlugin;
import org.apache.cordova.api.PluginResult;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

/**
 * This class echoes a string called from JavaScript.
 */
public class Echo extends CordovaPlugin {
    @Override
    public boolean execute(String action, JSONArray args, CallbackContext callbackContext)
    throws JSONException {

        if (action.equals("echo")) {
            String message = args.getString(0);
            if (message != null && message.length() > 0) {
                callbackContext.success(message);
            } else {
                callbackContext.error("Expected one non-empty string argument.");
            }
            return true;
        }
        return false;
    }

}
```

Extend the Cordova Plugin class

Implement execute method

Define and handle action

Plugin Example (HTML + JS Code)

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Cordova Plugin Test</title>
    <script type="text/javascript" src="cordova-2.0.0.js"></script>
    <script type="text/javascript">
    var EchoPlugin = {
        callNativeFunction: function (success, fail, resultType) {
            return Cordova.exec( success, fail, "sample.cordova.plugin.Echo", "echo", [resultType]);
        }
    };

    function callNativePlugin( returnSuccess ) {
        HelloPlugin.callNativeFunction( nativePluginResultHandler, nativePluginErrorHandler, returnSuccess );
    }
    function nativePluginResultHandler (result) {
        alert("SUCCESS: \r\n"+result );
    }
    function nativePluginErrorHandler (error) {
        alert("ERROR: \r\n"+error );
    }
    </script>
  </head>
  <body>
<body onload="onBodyLoad()">
<h1>Cordova Plugin Test</h1>
<button onclick="callNativePlugin('success');">Click to invoke the Native Plugin!</button>
</body>
```

Resources

* **Apache Cordova Website**
  http://cordova.apache.org/
* **Apache Cordova Documentation**
  http://docs.phonegap.com/en/2.2.0/index.html

* PhoneGap Day 2011 – IBM, PhoneGap and the Enterprise by *Bryce Curtis* [Aug 10, 2011]
  http://www.slideshare.net/drbac/phonegap-day-ibm-phonegap-and-the-enterprise (video)
* Andrew Trice's Blog
  http://www.tricedesigns.com/category/cordova/