POLITECNICO DI MILANO

# PhoneGap

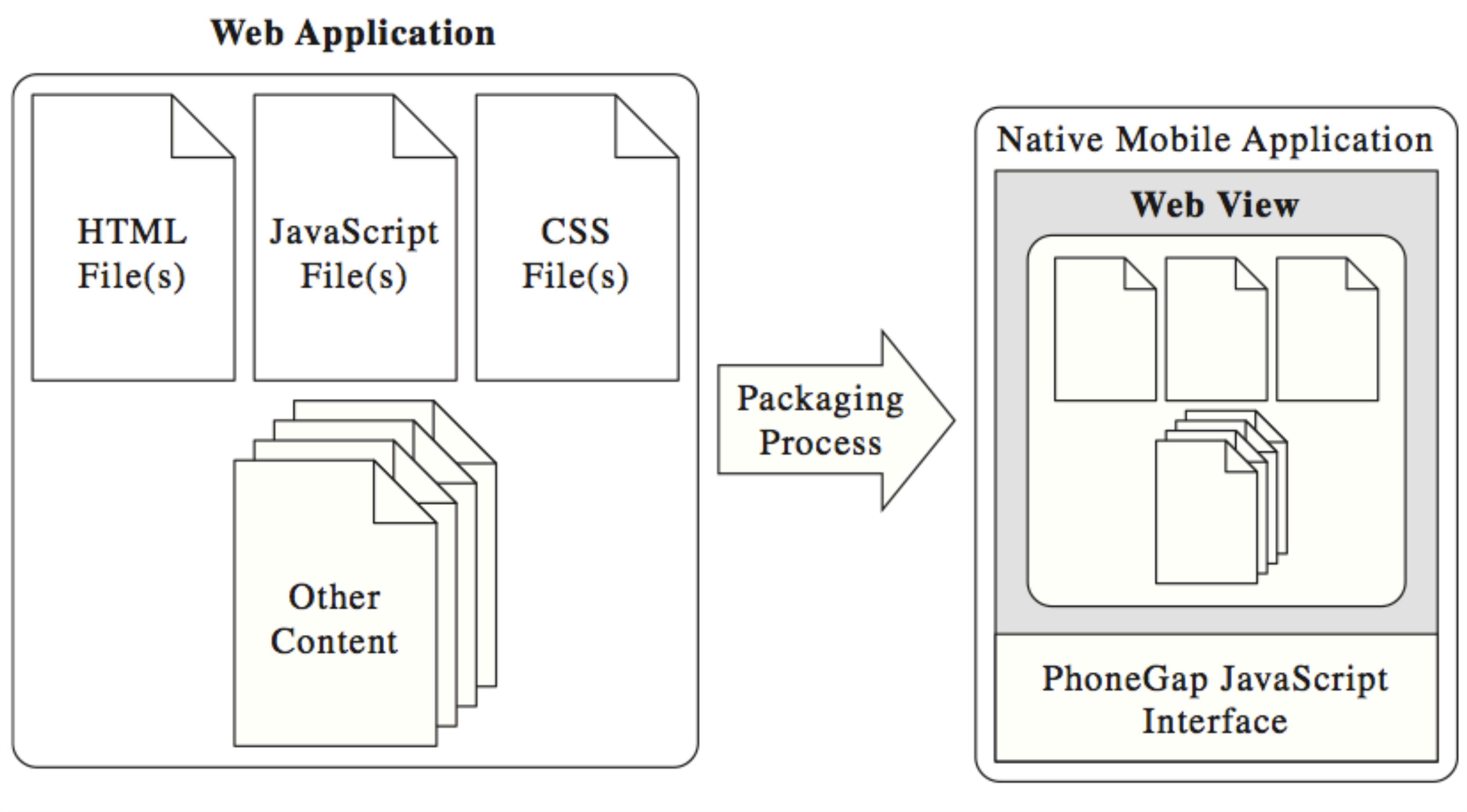
## Introduction to Mobile Application Development

# What is PhoneGap?

- Open source framework for building cross-platform native applications using standard web technologies
  - HyperText Markup Language (HTML)
  - Cascading Style Sheets (CSS)
  - JavaScript.

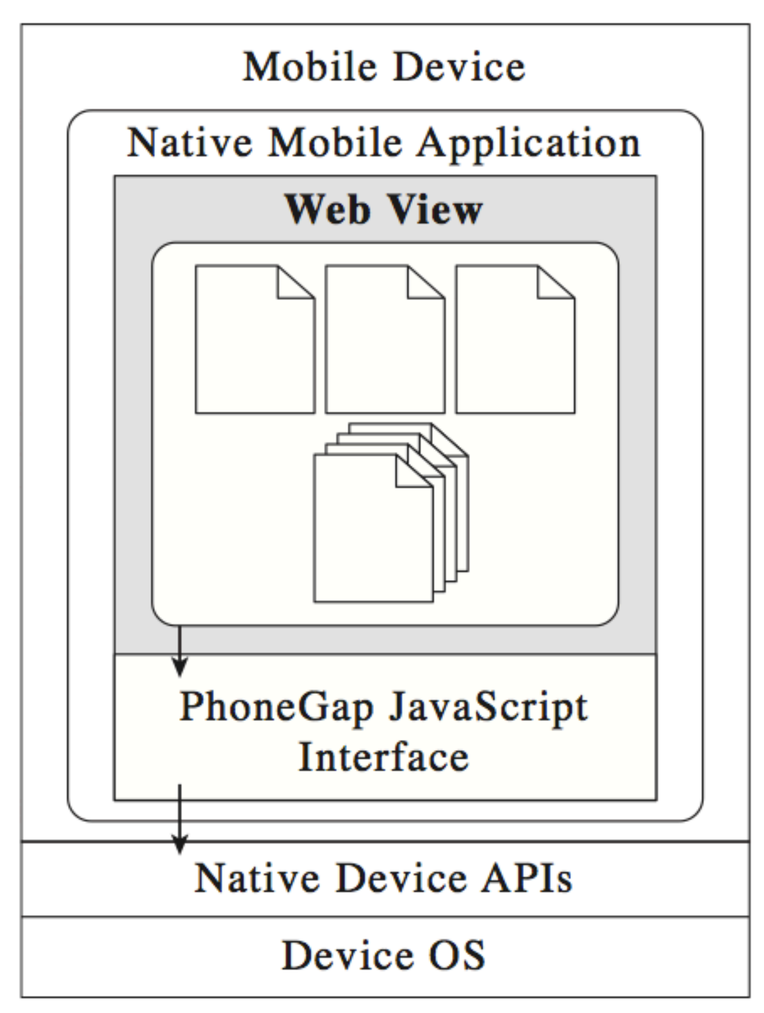
- Generates hybrid applications and supports several platforms
  - Apple IOS
  - Google Android
  - HP Palm/WebOS
  - Microsoft Windows Phone
  - Nokia Symbian
  - RIM
  - Samsung BADA

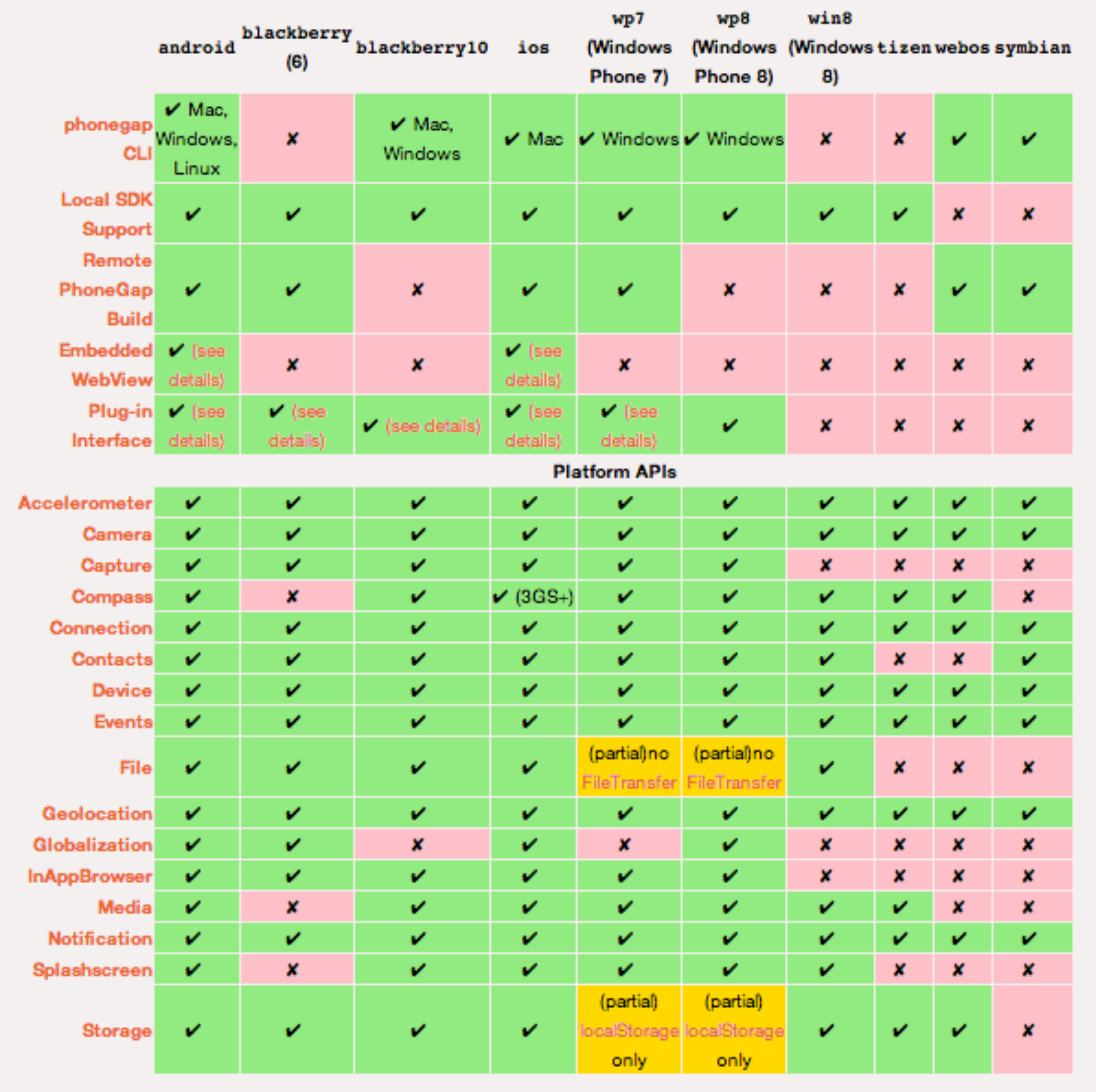
# How Does PhoneGap Work?



# How Does PhoneGap Work?

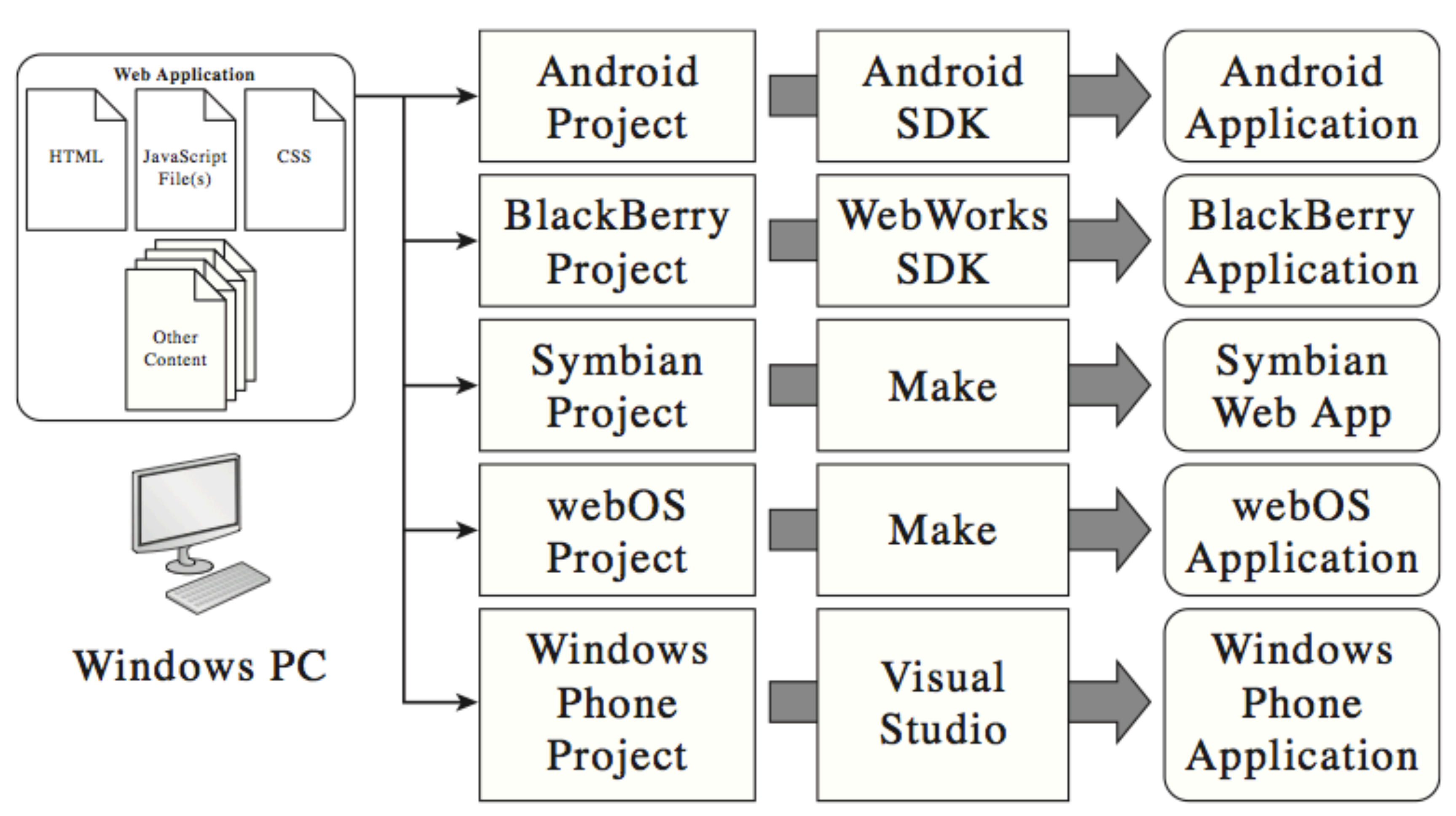
- Accelerometer
- Camera
- Capture
- Compass
- Connection
- Contacts
- Device
- Events
- File
- Geolocation
- Media
- Notification
- Storage



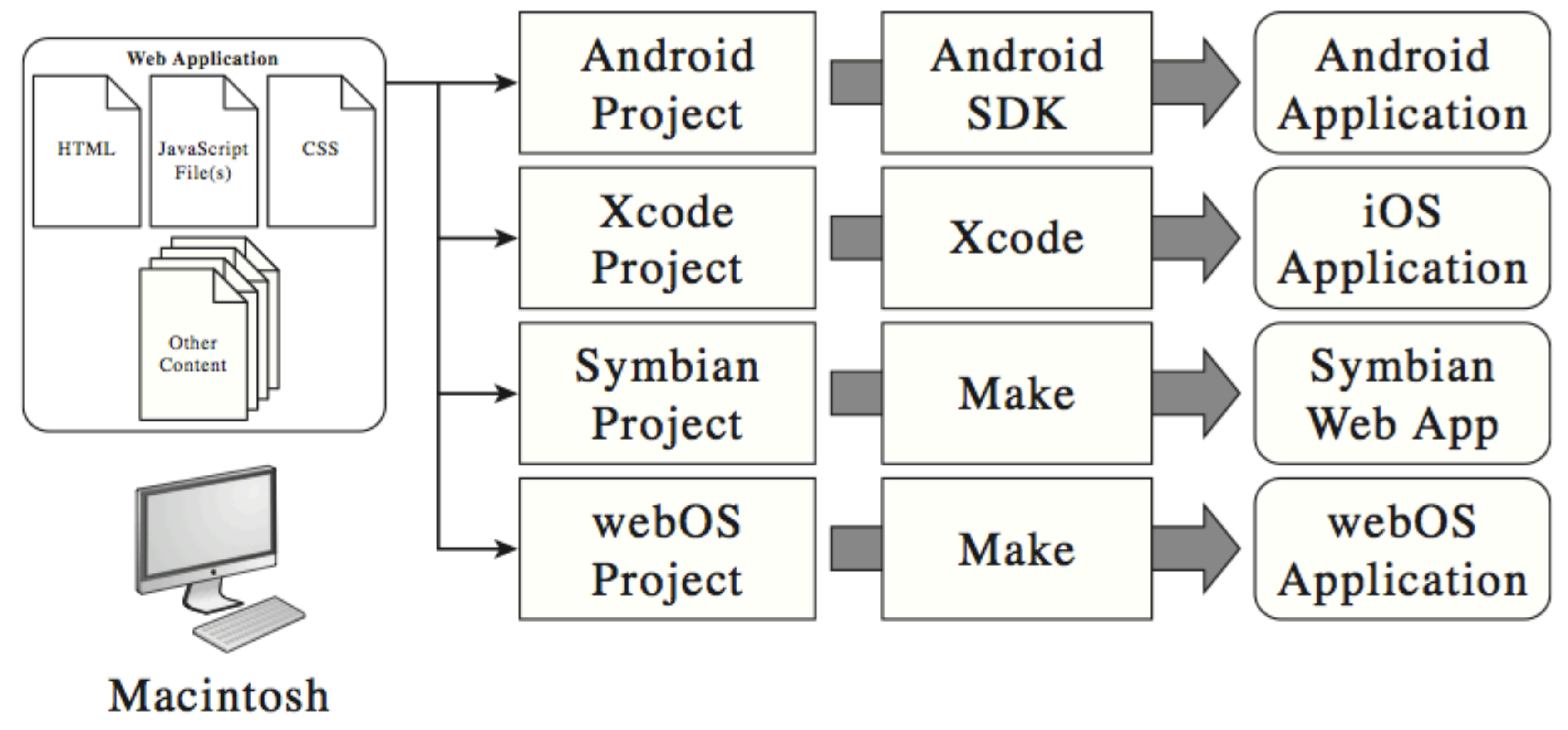


| | android | blackberry (6) | blackberry10 | ios | wp7 (Windows Phone 7) | wp8 (Windows Phone 8) | win8 (Windows 8) | tizen | webos | symbian |
|---|---|---|---|---|---|---|---|---|---|---|
| phonegap CLI | ✔ Mac, Windows, Linux | ✘ | ✔ Mac, Windows | ✔ Mac | ✔ Windows | ✔ Windows | ✘ | ✘ | ✔ | ✔ |
| Local SDK Support | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ |
| Remote PhoneGap Build | ✔ | ✔ | ✘ | ✔ | ✔ | ✘ | ✘ | ✘ | ✔ | ✔ |
| Embedded WebView | ✔ (see details) | ✘ | ✘ | ✔ (see details) | ✘ | ✘ | ✘ | ✘ | ✘ | ✘ |
| Plug-in Interface | ✔ (see details) | ✔ (see details) | ✔ (see details) | ✔ (see details) | ✔ (see details) | ✔ | ✘ | ✘ | ✘ | ✘ |
| **Platform APIs** | | | | | | | | | | |
| Accelerometer | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Camera | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Capture | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✘ | ✘ |
| Compass | ✔ | ✘ | ✔ | ✔ (3GS+) | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ |
| Connection | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Contacts | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✔ |
| Device | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Events | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| File | ✔ | ✔ | ✔ | ✔ | (partial)no FileTransfer | (partial)no FileTransfer | ✔ | ✘ | ✘ | ✘ |
| Geolocation | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Globalization | ✔ | ✔ | ✘ | ✔ | ✘ | ✔ | ✘ | ✘ | ✘ | ✘ |
| InAppBrowser | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✘ | ✘ |
| Media | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ |
| Notification | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Splashscreen | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✘ | ✘ |
| Storage | ✔ | ✔ | ✔ | ✔ | (partial) localStorage only | (partial) localStorage only | ✔ | ✔ | ✔ | ✘ |

# Building Process in Windows



# Building Process in Mac OSX



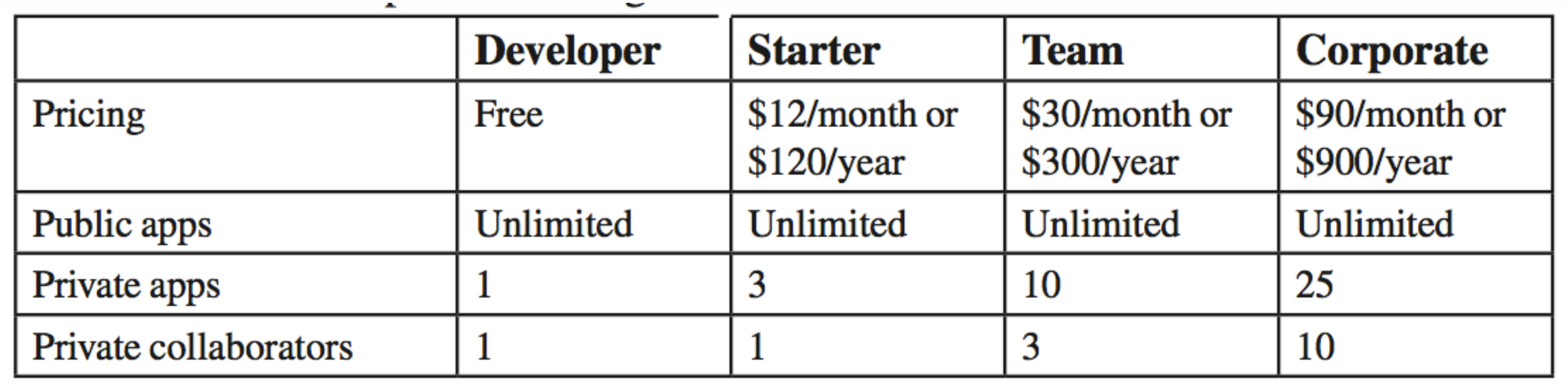
# The Good and the Limitations

- The Good
  - Transfer web apps to the mobile market easily
  - Enrich existing web apps with technologies available on mobile platforms (camera, geolocation, etc.)
  - Quick prototyping

- Any “Limitation”?
  - It is still a webapp and looks like a webapp
  - The look and feel is not native
  - Features and bug fixes for more popular platforms (such as Android and iPhone) get more attention while less popular plat- forms languish

# PhoneGap Build

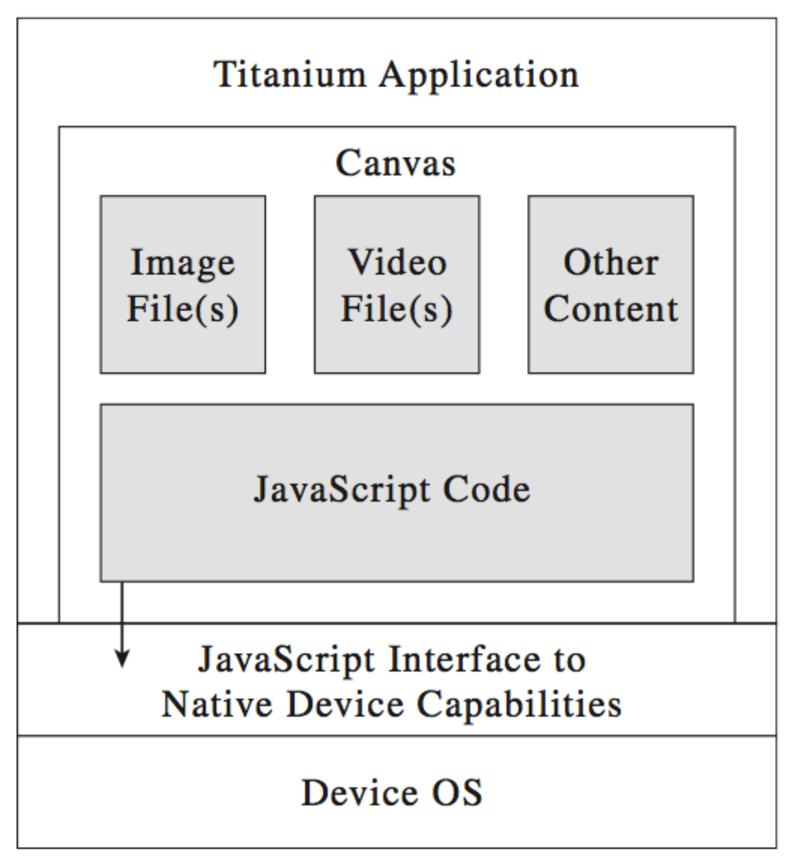
- Eliminate the need to install each platform’s SDK, as well as IDEs, build tools, simulators or emulators, and more
- Let developers build PhoneGap applications in the cloud
- PhoneGap Build currently uses a single application icon and a single splash screen image for all versions of the application (except for iOS)
- For iOS, PhoneGap Build still needs the application without the appropriate developer credentials/provisioning profiles



| | Developer | Starter | Team | Corporate |
|---|---|---|---|---|
| Pricing | Free | $12/month or $120/year | $30/month or $300/year | $90/month or $900/year |
| Public apps | Unlimited | Unlimited | Unlimited | Unlimited |
| Private apps | 1 | 3 | 10 | 25 |
| Private collaborators | 1 | 1 | 3 | 10 |

# Appcelerator Titanium

- Appcelerator Titanium
  - Works very similarly to PhoneGap in that developers build mobile applications using web technologies
  - The application's user interface and application logic are all coded entirely in JavaScript



# Let’s Install PhoneGap

- Install NodeJS
  - Go to http://nodejs.org/dist/
  - Download and install version v0.10.18

- Then install PhoneGap by running the following
  - sudo npm install -g phonegap

- If you work with IOS then also install the simulator by running
  - sudo npm install -g ios-sim

# First App

- Create the first app running the following commands
  - phonegap create example01
  - phonegap create example01 --name example01 --id net.pierlucalanzi.pge01
  - cd example01
- Run it for Android or IOS using
  - phonegap local run android
  - phonegap local run ios
- Command line interface
  - http://docs.phonegap.com/en/3.0.0/guide_cli_index.md.html

# PhoneGap Plugins

- Extend PhoneGap applications with additional functionality

- Even basic functionalities are now available as plugins (even the device information)

- Access to the device layer is typically available via a plugin

- List of standard plugins
  - http://docs.phonegap.com/en/3.0.0/guide_cli_index.md.html

# The First Example

- Create the app for the first lecture
  - phonegap create lec01 --name lec01 --id net.pierlucalanzi.lec01
  - phonegap local build android
  - phonegap local build ios
- The “app” is located in the www directory inside the project
- For IOS, open Xcode and load the project
- In Android, create a virtual device, the add the app
  - http://developer.android.com/tools/devices/managing-avds-cmdline.html
  - android list targets
  - android create avd -n <name> -t <targetID> [--skin WVGA800]
  - android create avd -n my_android1.5 -t 2 -p path/to/my/avd
  - android avd
  - emulator -avd <avd_name>
  - adb install lec01-debug.apk

# Device Information

```
var isPhoneGapReady = false;
var isPhoneGapConnected = false;
var isNetworkHighSpeed = false;

function device_information(id)
{
    var element = document.getElementById(id);
    element.innerHTML =
        'Device Name: '     + device.model     + '<br />' +
        'Device Cordova: '  + device.cordova   + '<br />' +
        'Device Platform: ' + device.platform  + '<br />' +
        'Device UUID: '     + device.uuid      + '<br />' +
        'Device Model: '    + device.model     + '<br />' +
        'Device Version: '  + device.version   + '<br />';
}
```

```
function network_information(id)
{
    var element = document.getElementById(id);
    if (isPhoneGapConnected && isNetworkHighSpeed)
    {
        element.innerHTML = 'CONNECTED AT HIGH SPEED';
    } else if (isPhoneGapConnected) {
        element.innerHTML = 'CONNECTED AT SLOW SPEED';
    } else {
        element.innerHTML = 'NOT CONNECTED';
    }
}
```

# Network Information

```
function detect_network() {
    if (isPhoneGapReady) {
        if (navigator.network.connection.type != Connection.NONE) {
            isPhoneGapConnected = true;

            switch (navigator.network.connection.type) {
                case Connection.UNKNOWN:
                case Connection.CELL_2G: isNetworkHighSpeed = false; break;
                default:
                    isNetworkHighSpeed = true;
                    break;
            }
        }
    }
}

        // check the network
        detect_network();

        // print the device properties
        device_information('device_properties');
        network_information('network_properties');

        // wait 5 seconds and then open the home page
        setTimeout(function() { window.open('home.html', '_self', false); }, 5000);
```

# Network Status Change

```
// attach events for online and offline detection
document.addEventListener("online", onOnline, false);
 document.addEventListener("offline", onOffline, false);

…

function onOnline() {isPhoneGapConnected = true; }
function onOffline() {isPhoneGapConnected  = false; }
```

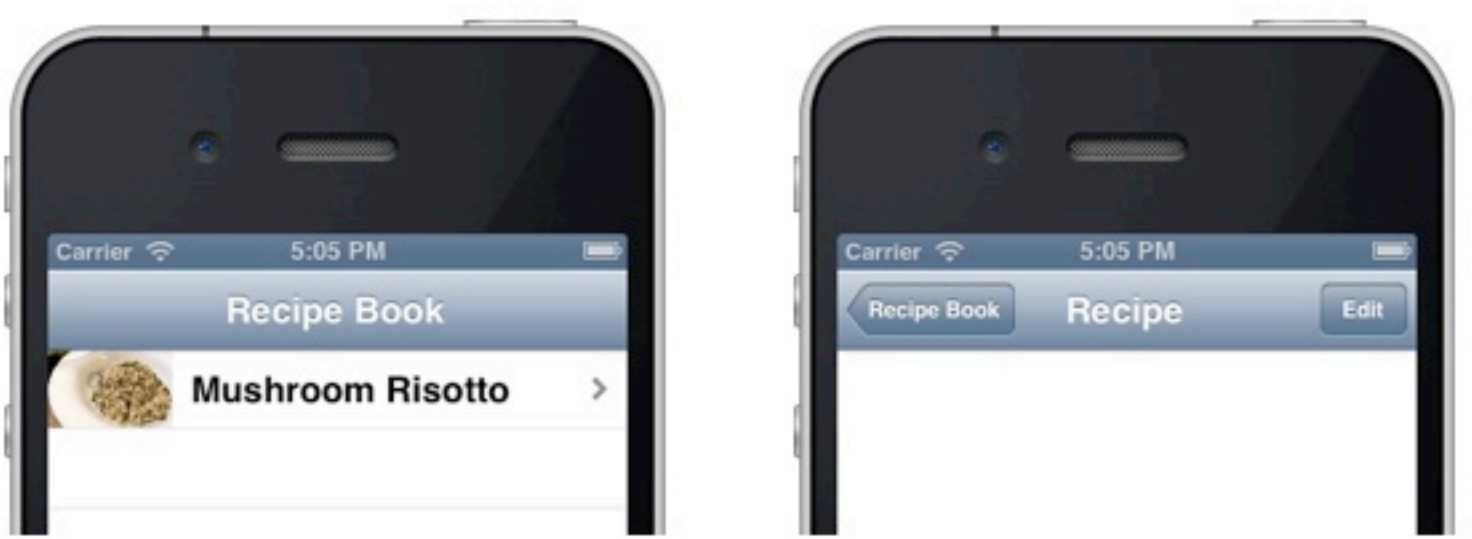
# jQuery and jQueryMobile

- jQuery
  - A JavaScript Library supporting, event handling, animation, Ajax interactions
  - It is CSS3 compliant and cross-browser
- jQuery Mobile
  - Free, open-source, JavaScript library that makes it much easier to develop web sites for mobile devices. (It is used in combination with the core jQuery library.)
  - jQuery Mobile stores multiple pages in a single HTML file, create dialog boxes, buttons, and navigation bars; format your pages without coding your own CSS; collapsible content blocks, and accordions.

# Header Bars

```
<div data-role="header">
  <h1 id="header">My Application</h1>
</div>

<div data-role="header">
  <a href="index.html" data-icon="delete">Cancel</a>
  <h1>Edit Contact</h1>
  <a href="index.html" data-icon="check">Save</a>
</div>
```

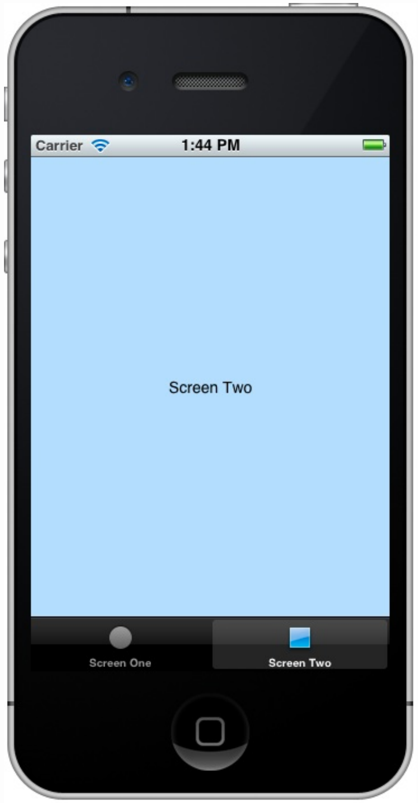


# Navigation Bar

```
<link rel="stylesheet" href="css/jquery.mobile-1.3.2.min.css" />
<script src="js/jquery-1.10.2.min.js"></script>
<script src="js/jquery.mobile-1.3.2.min.js"></script>

...

<div data-role="footer" data-position="fixed">
   <div data-role="navbar">
   <ul>
      <li><a href="home.html">Home</a></li>
      <li><a href="about.html"
             class="ui-btn-active">About</a></li>
   </ul>
   </div>
</div>
```

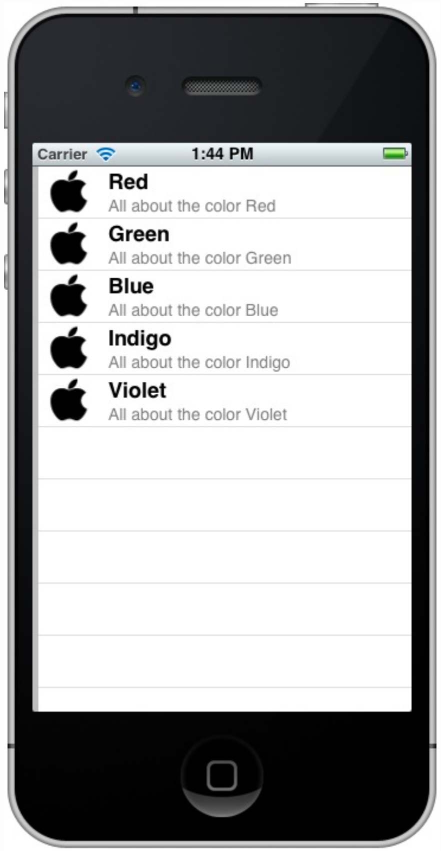


- Navigation Bar Options
  - http://api.jquerymobile.com/navbar/

# Table Views

```
<ul data-role="listview">
<li><h1>Farm</h1>
   <p>All the animals found in farms</p>
   <ul><li><a href="#">Cows</a></li>
       <li><a href="#">Chickens</a></li>
       <li><a href="#">Pigs</a></li></ul>
</li>
<li><h1>Wild<h1>
  <p>Those found in the wild</p>
  <ul><li><a href="#">Giraffes</a></li>
      <li><a href="#">Lions</a></li>
      <li><a href="#">Tigers</a></li></ul>
</li>
...

</ul>
```


Carrier 1:44 PM

Red
All about the color Red

Green
All about the color Green

Blue
All about the color Blue

Indigo
All about the color Indigo

Violet
All about the color Violet

- Examples
  - http://jquerymobile.com/demos/1.1.1/docs/lists/lists-nested.html

# Application Structure (One File)

```
<div data-role="page" id="home”>
  <div data-role="header”>
     ...
     <h1>Home</h1>
     ...
  </div>

  <div data-role="content">
  Hello and welcome to my home ...
  </div>
   <div data-role="navbar">
   <ul>
     <li><a href="#home" data-transition="none" data-icon="home">Home</a></li>
     <li><a href="#compass" data-transition="none" data-icon="home">Home</a></li>     ...
   </ul>
   </div>
</div>

<div data-role="page" id=”compass">
<!– same structure -->
</div>
```

# Buttons and Other Stuff

- JQueryMobile has a wide variety of form elements
  - http://jquerymobile.com/demos/1.0/docs/forms/docs-forms.html

- Buttons

  -
    ```
    <button id=”networkstatus" data-icon="check">Network is on</button>
    if(navigator.network.connection.type == Connection.NONE){
    $("#home_network_button").text('No Internet Access’)
    .attr("data-icon", "delete”)
    .button('refresh');
    }
    ```

  -
    ```
    <button id=”do_stuff">clear</button>
    $("#home_clearstorage_button").live('click', function(){…});
    ```

# PhoneGap Events

- deviceready
- pause
- resume
- online
- offline
- backbutton
- menubutton
- searchbutton

# Pause and Resume

- Determines what the application will do when it is paused and resumed. It typically involves saving temporary data, etc.

```
//Start time variable
var startTime, endTime;
//PauseInfo variable
var pi;
//FirstTime variable
var firstTime;

// when ready
//Add our Pause event listener
document.addEventListener("pause", processPause, false);
//Add our Resume event listener
document.addEventListener("resume", processResume, false);
//Get a handle to the pauseInfo page element
pi = document.getElementById("pauseInfo");
```

# Pause and Resume

```
function processPause() {
  //Clear the previous counter
  pi.innerHTML = "Application paused.";
  //Set startTime to the current date/time
  startTime = new Date();
}

function processResume() {
  //alert("resume");
  //We want to skip the first time this fires
  if(firstTime == true) {
    //Clear our firstTime variable
    firstTime = false;
    pi.innerHTML = "Skipping first Resume.";
} else {
    //Get the current date
    endTime = new Date();
    timeDiff = (endTime - startTime) / 1000;
   //Update the screen
pi.innerHTML = "Paused for " + timeDiff + " seconds.”; } }
```

# Pause and Resume Work Differently in Different OS

```
function onDeviceReady() {
    alert("onDeviceReady");
    pName = device.platform;
    // in Android and Blackberry resumes fires also
    // when the application starts
    if((pName == "Android") || (pName == "3.0.0.100")) {
        firstTime = true;
    } else {
        firstTime = false;
    }

    //Add our Pause event listener
    document.addEventListener("pause", processPause, false);
    //Add our Resume event listener
    document.addEventListener("resume", processResume, false);
    //Get a handle to the pauseInfo page element
    pi = document.getElementById("pauseInfo");
}
```

# Accessing Device Specific Hardware

- Add the plugin and check that the code specific is included into the platform-specific code

- Wait for the device to be ready
  - document.addEventListener("deviceready", onDeviceReady, false);

- The init function also inits the hardware specific routines
  - navigator.compass.getCurrentHeading(onSuccess, onError);
  - var compassOptions = { frequency: 1000 };
    var watchID = navigator.compass.watchHeading(onSuccess, onError, compassOptions);

# Online & Offline

```
function isOnline() {
    //alert("isOnline");
    var d = new Date();
    $('#networkInfo').prepend("Online: " + d.toLocaleString() + "<br />");
  }

  function isOffline() {
    //alert("isOffline");
    var d = new Date();
    $('#networkInfo').prepend("Offline: " + d.toLocaleString() + "<br />");
  }

 <body onload="onBodyLoad()">
    <h1>Network Tracker</h1>
    <p id="networkInfo"></p>
  </body>
```

# Compass (Fixed) - Javascript

```
 <script type="text/javascript">
         document.addEventListener("deviceready", onDeviceReady, false);

         function onDeviceReady() {
            navigator.compass.getCurrentHeading(onSuccess, onError);
        }

         function onSuccess(heading) {
            alert('Heading: ' + heading.magneticHeading);
            var angle = heading.magneticHeading;
            $('#compass').css('transform','rotate(' + angle + 'deg)');
         }

        // onError: Failed to get the heading
         function onError(compassError) {
            alert('Compass Error: ' + compassError.code);
        }
</script>
```

# Compass (Fixed) - HTML

```
<div id="page1" data-role="page">
              <header data-role="header" data-position="fixed">
                      <h1>Compass</h1>
              </header>
              <section data-role="content" style="padding: 10px 0;">
                      <img id="compass"
                                class="Center"
                                width="80%"
                                height="80%"
                                src="images/compass320.png" />
              </section>
      </div>
```

# Storage (no plugin required)

- Local Associative Storage
  - value = window.localStorage.getItem("key");
  - window.localStorage.setItem(’’key", value);
  - window.localStorage.clear();

- SQL Database
  - PhoneGap can also read and write to structured SQL tables

# SQL Storage

- db = window.openDatabase(db_name, db_version, db_display_name, db_size);

    - db_name: The name of the database (the file in the device memory)
    - db_version: The version number for the database. An application can query this version number and upgrade the database schema as needed using the changeVersion method of the database object.
    - db_display_name: The display name for the database.
    - db_size: The amount of space allocated for the database in bytes. When allocating space, keep in mind that mobile devices may have limitations on the size of databases they can support, so allocate only the amount of space you think the application will need.

# SQL Storage

- SQL statements are wrapped inside transactions
  - db.transaction(createTable, onTxError, onTxSuccess);

- Error

```
function onTxError(tx, err) {
  var msgText;
  if(err) {
    //Tell the user what happened
    msgText = "TX: " + err.message + " (" + err.code + ")";
  } else {
    msgText = "TX: Unknown error";
  }
  console.error(msgText);
  alert(msgText);
}
```

# SQL Storage

-
  ```
  function onTxSuccess() { console.log("TX: success"); }
  ```

-
  ```
  function createTable(tx) {
  var sqlStr = 'CREATE TABLE IF NOT EXISTS MILEAGE
  (tripDate INT, miles INT, notes TEXT)’;
  console.log(sqlStr);
  tx.executeSql(sqlStr, [], onSqlSuccess, onSqlError);
  }
  ```

# SQL Storage

```
function onSqlSuccess(tx, res) {
  console.log("SQL: success");
  if(res) {
    console.log(res);
  }
}

function onSqlError(tx, err) {
  console.log("Entering onSqlError");
  var msgText;
  if(err) {
    msgText = "SQL: " + err.message + " (" + err.code + ")";
  } else {
    msgText = "SQL: Unknown error";
  }
  console.error(msgText);
  alert(msgText);
  console.log("Leaving onSqlError");
}
```

# SQL Statements

```
function onSqlSuccess(tx, res) {
  if(res) {
    console.log("Insert ID: " + res.insertID);
    console.log("Row affected: " + res.rowAffected);
    if (res.rows) {
       var len = res.rows.length;
       if(len > 0) {
         for(var i = 0; i < len; i++)
          { //Do something with the row data }
       } else {
          alert("No records processed.");
       }
     }

   } else { alert("No results returned."); } }
```

# File Storage

- Applications can have a persistent or temporary or temporary, but must request it to the underlying OS
  - window.requestFileSystem(fileSystemConstant, sandboxSize, onSuccessFunction, onErrorFunction);
  - window.requestFileSystem(LocalFileSystem.TEMPORARY, 5 * 1024 * 1024, onSuccessFunction, onErrorFunction);
- As usual, there is a success function (you have the access)

  -
    ```
    function onSuccessFunction(fs) {
       alert("Accessing " + fs.name + " storage (" + fs.root.fullPath + ")");
       // do stuff here
    }
    ```

  -
    ```
    function onFileError(e) {
       var msgText;
       switch(e.code) { … };
       alert( …);
    ```

# File Storage – Accessing Dirs

```
function onGetFileSystemSuccess(fs){
  alert("Accessing " + fs.name + " storage (" + fs.root.fullPath + ")");
  //Create a directory reader we'll use to list the files in
  //the directory var dr = fs.root.createReader();
  // Get a list of all the entries in the directory
  dr.readEntries(onDirReaderSuccess, onFileError);
}

function onDirReaderSuccess(dirEntries) {
  len = dirEntries.len;
  for(i=0; i<len; i++)
  {
     // do stuff with dirEntries[i].name, dirEntries[i].isDirectory etc.
  }
}
```

# References

- http://docs.phonegap.com/en/edge/index.html
- http://docs.phonegap.com/en/3.0.0/guide_cli_index.md.html