By Stephen Cavell, Kerry Ellwanger, and Jack Livingston
History

• PhoneGap created in 2009 by startup Nitobi.
• Open source way to access the “native” environment through an embedded WebView in a native app.
• Goal: ability to build the majority of a mobile app with “pure web technologies” like HTML5, CSS, and Javascript.
• Purchased in 2011 by Adobe.
• Open source core donated to the Apache Software Foundation under the name Cordova.
How Cordova & PhoneGap Work

- Users write HTML, JS, CSS as they would a website.
- Can include plugins that handle native processes.
- Cordova & PhoneGap package resources into an app that interacts with native OS information.
Cordova vs. PhoneGap

- Analogy: Cordova is to Webkit as Phonegap is to Chrome.
- Differences between Cordova and PhoneGap were originally minimal, but Adobe has been building a proprietary set of services around the PhoneGap ecosystem and use the PhoneGap distribution of Cordova.
- When starting a hybrid app project:
  - Can use Cordova proper.
  - Enter into Adobe’s ecosystem and use the PhoneGap distribution of Cordova.
Benefits

• Developers can use an often more familiar skill set.
• Native Capabilities accessible via PhoneGap.
• Clients can maintain a single codebase and deploy it through multiple app stores.
## Platforms and Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>iPhone / iPhone 3G</th>
<th>iPhone 3GS and newer</th>
<th>Android</th>
<th>Blackberry OS 6.0+</th>
<th>Blackberry 10</th>
<th>Windows Phone 8</th>
<th>Ubuntu</th>
<th>Firefox OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Camera</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Compass</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Contacts</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>File</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Geolocation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Media</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Network</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Notification (Alert)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Notification (Sound)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Notification (Vibration)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Storage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
How do we utilize these platform features?
Plug-Ins

• Bridges functionality between the WebView powering a Cordova/PhoneGap application and the native platform the application is running on.
• Plugins use a single JavaScript interface across all platforms.
• Every plugin must use a cordova.exec function to communicate between the Cordova Javascript and native environments.

```javascript
cordova.exec(function(winParam) {}, function(error) {}, "service", "action", ["firstArgument", "secondArgument", 42, false]);
```
Downsides and Limitations

• Difficulty getting apps approved
• Lacks built in UI tools
• Performance
• Full functionality does not exist
Application Approval in iOS

Reasons Apple might reject an app:

- Look and feel
- Performance issues
- Downloads a significant amount of data
- Downloads any code in a way that an external resource can change the functionality or look of the app
- Accepting payment outside of the appstore
- App contains bugs
- App isn’t relevant to the general population
Performance

- Javascript running in an interpreter is going to be slower than native code
- Some features such as the use of “onclick()” have been reported to cause noticeable latency
- Loading larger apps can take a significant amount of time
When to Use PhoneGap

- Cross Platform is important
- UI/.Graphics are minimal so performance isn’t as much of an issue
- Developers are more comfortable with html, css, and javascript
Cross-platform work flow vs. Platform-centered work flow

- CPW: Utilizes the Cordova Command Line Interface. Can build a project for multiple platforms at once.
- PCW: Work flow more centered on a specific app. Rather than operating through a terminal development largely takes place in a specific SDK. Allows more customization and inclusion of native application components.
Installing Cordova/Using

• Need Node.js as well as a git client (CLI Behind the scenes)

• Use npm to install
  • $ sudo npm install –g cordova

• To use Cordova for specific platforms, you will need relevant SDK’s (Software Development Toolkit).

• For IOS this requires Xcode.
Creating an App

- Navigate to desired directory
  - `$ cordova create hello com.example.hello HelloWorld`
- First (required): specifies directory to be created for project
- Second: reverse domain-style identifier
  - Default: `io.cordova.hellocordova`
- Third: application’s display title
  - Default: `HelloCordova`
Adding platforms

• You can add platforms to your project while in your project directory
• Before building a project, you must add your target platforms.
  • `$ cordova platform add ios`
  • `$ cordova platform add android`
  • `$ cordova platform add amazon-fireos`
• Can also remove platforms:
  • `$ cordova platform remove blackberry10`
  • `$ cordova platform rm android`
  • `$ cordova platforms ls`
Building your project

• Build
  • $ cordova build
  • $ cordova build ios
  • $ cordova emulate ios

• Prepar/Compile
  • $ cordova prepare
  • $ cordova compile
Adding Plugins

• A newly created cordova project has no plugins
• Plugin search
  • $ cordova plugin search bar code
  • com.phonegap.plugins.barcodescanner
• Plugin add – adds plugins as appropriate for each platform
  • $ cordova plugin add org.apache.cordova.device
  • $ cordova plugin add https://github.com/apache/cordova-plugin-console.git
• Plugin list
  • $ cordova plugin ls
Cool links!

- Documentation
  - http://cordova.apache.org/docs/en/3.6.0/

- Featured Phonegap apps
  - http://phonegap.com/app/
Thank you!